Aalto University
School of Engineering

Cesare Palestini

Engine crankshaft torsional vibration analysis for anomalies detection

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 31.5.2018
Supervisor: Professor Petri Kuosmanen
Advisors: Dr Irene Gallici and Dr Juho Könnö

Aalto University
School of Engineering

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Cesare Palestini

Title Engine crankshaft torsional vibration analysis for anomalies detection

Degree programme Mechanical Engineering

Major Master's Programme in Mechanical Engineering **Code of major** ENG25

Supervisor Professor Petri Kuosmanen

Advisors Dr Irene Gallici, Dr Juho Könnö

Date 31.5.2018 **Number of pages** 61+4 **Language** English

Abstract

Reliability is a key factor in medium speed engines. Since they are employed in power plants and vessels, there is the constant need to increase the accuracy of their anomaly detection loops. Here I investigated a novel idea aimed at decreasing the number of sensors deployed. The goal was to understand whether using only one speed sensor, located at the flywheel, is enough to detect and possibly classify different abnormal operating conditions.

A torsional vibration model of a V20 spark-ignited gas engine was developed and validated with field data. The model was then used to simulate normal, heavy knock, misfire and overpressure conditions. Subsequently, the speed signals of the crankshaft, measured at the flywheel, were analysed in the time- and frequency-domains, in order to identify possible patterns that could be linked to the four scenarios. The spectra of the signals were then processed with two machine-learning approaches: pattern recognition and a neural network.

Both approaches showed accuracies up to 97% in the classification of the scenarios. Amongst the pattern recognition algorithms, the Ensemble with Subspace kNN proved to be the most accurate one, with a remarkably low rate of false negatives (<1%). The neural network had a slightly lower accuracy in the binary identification between normal and abnormal situations, but it performed better in classifying the different anomalies. The results are promising. They indicate that, with some further improvement, the method tested and developed in this thesis could be a reliable alternative to the safety systems currently in use in engines.

Keywords torsional vibration, speed measurement, multi-body dynamics, anomalies detection, machine learning, pattern recognition, artificial neural network, data analytics, heavy knock, overpressure, misfire

Acknowledgements

This master's thesis is part of a research project funded by Wärtsilä Finland Oy and it aims to investigate new horizons in the field of reliability engineering. It was a fantastic opportunity for me to discover a fascinating world populated by huge engines, learn new things in the field of the systems simulation and the data analysis, and to grow professionally and as a person.

I want to thank my supervisor Professor Petri Kuosmanen and my advisors Irene Gallici and Juho Könnö for the guidance received. I also thank the owner and the manager of the research project, Andreas Hjort and Fabio Mauceri, for their support.

In the time spent working on my thesis I had the possibility to meet extremely skilled and interesting colleagues that shared their knowledge with me. Among them, I want to cite Harry Särs for the insights on engines technology and the logistics assistance, Jan Kaas and Markus Björkvall for their help with the system simulation, Andrea Greco for his coaching on the data analysis. Last but not least, Jari Hyvönen for sharing his expertise. Without their precious hints this work would not have been possible.
Finally, I thank also Håkan Nynäs for the pleasant moments spent together with me and Harry talking about technology and funny stories of the past times.

I also want to take this opportunity to thank all the people that in these years in Finland supported me in my master studies: my family and my life partner Maria, the old and the new friends, the Wasa fellas.

Otaniemi, 31.5.2018

Cesare Palestini

Contents

Symbols and abbreviations

Symbols

$\underline{z}_i$	incremental stage variable
$\underline{\hat{z}}_i$	approximation of the incremental stage variable
$\underline{\varepsilon}_i$	truncation error
h	step size
$\underline{y}$	vector containing unknown variables
f	frequency

Operators

$\dot{.}$	derivative with respect to time

Abbreviations

LNG	liquefied natural gas
IEC	International Electrotechnical Commission
SIL	safety integrity level
MTBF	mean time between failures
SI	spark-ignition
SG	spark-ignited gas
BMEP	brake mean effective pressure
KI	knock intensity
GUI	graphical user interface
FEM	finite element method
CAD	computer aided design
CA	crankshaft angle
LPF	low pass filter
FFT	fast Fourier transform
ML	machine learning
PRA	pattern recognition algorithm
ANN	artificial neural network
TDC	top dead center
kNN	k nearest neighbors
SVM	support vector machines
ROC	receiver operating characteristic
FPR	false positive rate
TPR	true positive rate

1 Introduction

1.1 Background

Large combustion engines are the main source of energy in ships and certain types of power plants. Wärtsilä is a leading company in the development and production of these engines, which can use different fuels, from heavy fuel to LNG. It operates globally and offers tailored solutions for both marine and energy applications.
For energy production, the standard set-up usually consists in an engine connected to a generator by means of a flexible coupling, and the reliability of these systems is one of the key concerns that must be addressed.

1.2 Research problem

When operating large engines, it is imperative to minimize the risk of failure, since it can lead to additional costs due to repair, and cause temporary unwanted shut-downs and potentially harmful situations. Many customers require the manufacturers of the engines to equip their installations with safety systems that satisfy specific safety standards. In this case study, the standard involved is defined by the International Electrotechnical Commission as IEC 61508 and concerns the functional safety of electrical/electronic/programmable electronic safety-related systems. This standard defines four levels of risk reduction called Safety Integrity Level (SIL). The levels rank the different control systems, according to criteria of failure rate, redundancy, likelihood of failure, differentiation and criticality of the potentially harmful situation in case of failure. They range from 1 (the lowest) to 4 (the highest).

1.3 Scope

There are several events that can be harmful for an engine, and they should be prevented or at least detected as soon as possible, to avoid damages to the engine or explosions.
For spark ignition engines, one of these events is engine knock. Engine knock is defined by Wang, Liu and Reitz as "the name given to the noise associated with the auto-ignition of a portion of the fuel-air mixture ahead of the propagating flame front" in the combustion chamber (Wang, Liu, and D. Reitz 2017). It can damage the piston rings, cause seizure, piston and cylinder head overheating and, in the most extreme cases, lead to an explosion and to non-repairable damages. To detect the knock, the most common method consists in using one or several piezoelectric sensors called "knock sensors", located on the cylinder heads or on the engine block. These sensors are effective but their failure rate is too high to comply with the strict requirements of the SIL-2 standard.
Another potentially harmful situation for highly boosted engines is cylinder overpressure. In these engines, the pressure at the inlet is high due to the turbocharging. It may occur, for example in case of inlet valve failure, that the amount of fuel-air mixture introduced into the combustion chamber is above the design limits. The

consequence is that the combustion generates a peak pressure that is much higher compared to the other cylinders, and this may lead to a structural damage.
A third phenomenon studied in this work is engine misfire. Misfire occurs when the air-fuel mixture cannot ignite inside the combustion chamber, and thus there is no power generation in the cycle and unburnt fuel is released in the exhaust gases. Misfire does not jeopardize the engine integrity but should still be avoided, as it increases the emission of pollutants in the atmosphere and the fuel consumption, and introduces power unbalance between the cylinders.

1.4 Aim of the work

This work aims at investigating whether it is possible to detect ab-normal situations like the ones mentioned in the previous paragraph only by analyzing the speed fluctuations of the crankshaft. Such an approach would reduce drastically the minimum number of necessary sensors, consequently reducing the risk of failure. Moreover, speed sensors generally have a higher mean time between failures (MTBF) compared to knock sensors, which is necessary to fulfil the SIL requirements. On the other hand, analyzing speed fluctuations of the crankshaft also has disadvantages over a cylinder-by-cylinder monitoring system. The use of a limited number of sensors located on the flywheel or, more in general, on the crankshaft, does not allow the identification of the faulty cylinder. Tailored countermeasures, such as modifying the fuel-to-air ratio or the cam timings only in the cylinder where the knock is taking place, cannot be adopted. Another limitation is the difficulty in detecting small speed variations, due to the huge inertia that originates from the engine components and the torsional vibration damper that is generally located in the free end of the engine. As a consequence, only the more severe levels of knocking are detectable. Also, from the torsional point of view, the crankshaft can be seen as an ensemble of torsional springs linked together, with several excitation points (the throws) interposed in between. In a V20 engine the crankshaft assembly is more than 7.5 meters long and the vibrations can propagate inside the shaft in complex modes. The detection from the far ends can present noise or be difficult if the excitation comes from the opposite end with respect to the measurement point.

1.5 Methods

Our analysis of the torsional vibrations is focussed on a V20 spark ignition gas engine part of the Wärtsilä portfolio. This decision comes from the preference of focusing on the worst-case scenario, due to the complexity of the crankshaft and the amount of excitation points involved, and from the advantage of studying the most common engine type found in power plants. The V20 spark-ignited engine has 20 cylinders in a 50° angle and 40 cams acting on the cam-shafts, all of them generating vibrations at the same time. The software GT-SUITE was used to perform the simulations, which involve all the main rotating components of the engine, the coupling and the generator. The inputs for the model are taken directly from a prototype engine. After a validation phase, we simulated different abnormal scenarios. The output of

the model is then processed using pattern recognition and neural network algorithms. Finally, the results are collected and commented.

2 Theory

One of the key aspects in the development of modern engines is to increase the overall efficiency of the system. To do so, one of the most effective techniques is to increase the brake mean effective pressure (BMEP). This value is a parameter that relates the torque generated by the engine to the engine displacement and thus allows easy comparison between different systems. There are several techniques to achieve higher BMEP values, two of them comprehending higher compression ratios and turbocharging.

2.1 Engine knock

In spark-ignition (SI) combustion engines, one of the biggest problems that limits the possibility to increase the efficiency is the knock tendency. The engine knock is defined as an unwanted self-ignition of a portion of the fuel-air mixture in the combustion chamber. This detonation is harmful for the engine and, to some extent, it should be avoided (Heywood 1988). Engine knocking can be caused by several factors. Some of them are poor fuel quality, presence of carbon deposits on the surface of the combustion chamber, excessive temperature and high compression ratio.
Low quality fuels, when exposed to high pressure and temperature, are generally prone to auto-combustion, and their performance is assessed using a parameter called octane number which is an index of the fuel ability to resist knock. In SI engines, a self combustion does not allow to control the timing and could lead to decreased efficiency. Furthermore, if the detonation occurs close to the metal surface of the combustion chamber, this could affect the material properties and the efficiency of the lubricating film.
Carbon deposits inside the combustion chamber can be trigger points for a self ignition as they generally represent localized hot spots (Zbigniew 2014). The ignition starting from these deposits creates a pressure front that travels across the combustion chamber in contrast with the one generated by the spark plug.
The temperature inside the combustion chamber influences the likelihood of a self ignition too. This is even more true in case of a lean combustion where the lack of unburnt fuel does not reduce the temperature of the combustion. This would happen because the unburnt fuel does not contribute actively to the combustion but represents a matter that absorbs energy from the combustion, changing also its state in case of liquid fuel.
Similarly to high temperature, also a high compression ratio increases the risk to have knocking in the engine. This happens especially in case of high boost because the pressure at the beginning of the compression phase is already higher than the atmospheric pressure. Nevertheless, to increase the efficiency of the engine, it is common and good practice to recover of part of the unused energy in the exhaust gases and convert it into extra intake pressure to the engine by using a turbocharging system. The name “knock” originates from the peculiar sound associated with this phenomenon. The frequency of this sound is of function of the bore size and the shape

of the combustion chamber. Since under the definition of engine knock there are several trigger factors and ways in which the phenomenon develops, in the scientific literature there are several formulas available to estimate the knock frequency. Some of them take into account only the bore, some others also other factors, such as the temperature and the type of chemicals present in the combustion, but in general a linear approximation based on observed data gives reliable results if applied to the same type of engines. For Wärtsilä gas engines, for example, it has been observed that the equation:

$$\frac{952\ [m/s]}{2 * d\ [m]} = knock frequency\ [Hz] \tag{1}$$

gives a good estimation of the expected knock frequency. If we apply it to the W31 engine, which has a bore of 0.31 m, the results is:

$$\frac{952\ m/s}{2 * 0.31\ m} \approx 1535\ Hz \tag{2}$$

The result can be verified observing figure 1 where four different measured knock curves are represented in function of the crank angle and their constituent curves analyzed through a Fast Fourier transform analysis. The pressure waves generated

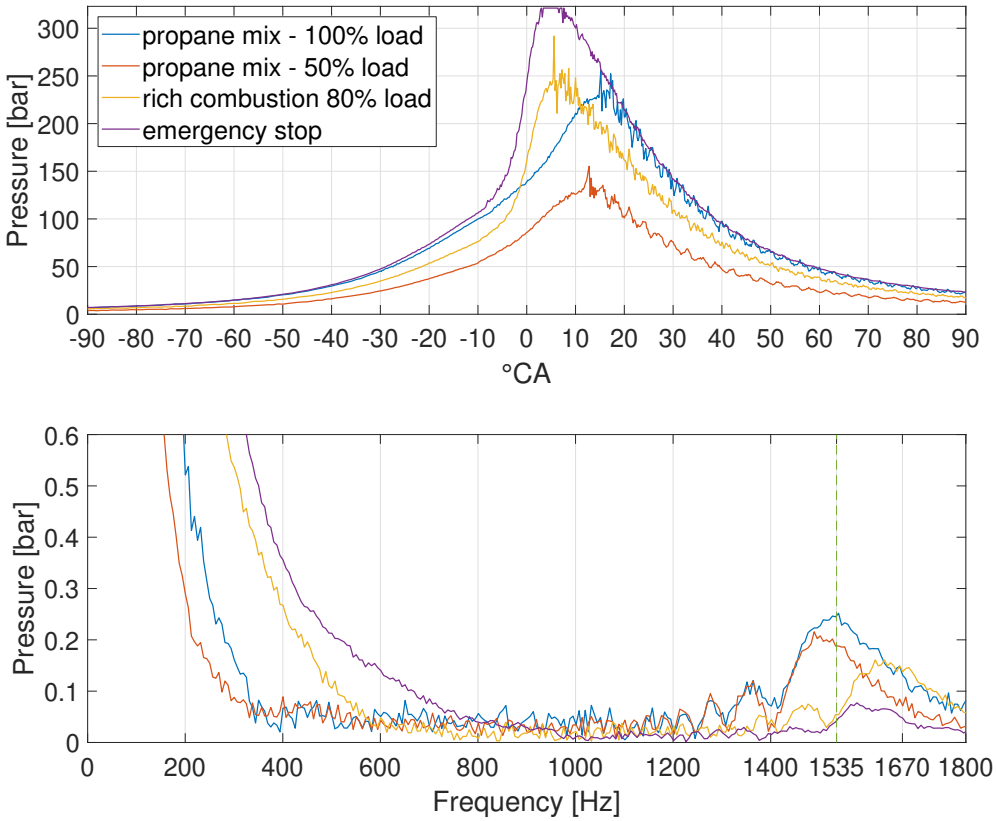

Figure 1: Knock pressure curves and frequency analysis

inside the combustion chamber during the knocking develop in several mode shapes; the main ones are illustrated in figure 2 where the CH emissions during the combustion are highlighted through a high-speed chemiluminescence imaging technique (He et al. 2015). The most relevant resonant mode is the first one, where the pressure wave

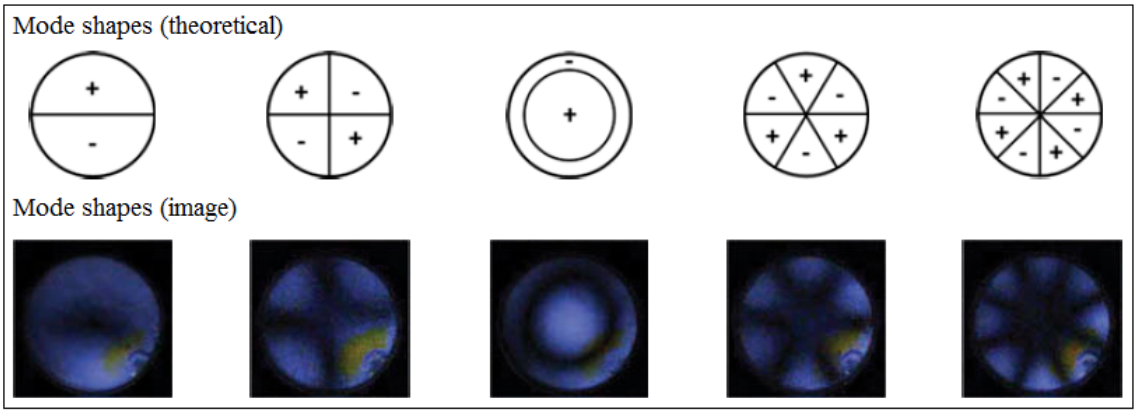

Figure 2: View of the first 5 mode shapes from the cylinder top (He et al. 2015)

travels horizontally inside the combustion chamber. The frequency of that horizontal oscillation corresponds to the characteristic knock frequency that is commonly sensed by means of piezoelectric sensors and possible to spot in the human audible range. Engine knock can be divided in two main groups: conventional knock and super-knock (Wang, Liu, Song, et al. 2014).

The conventional knock arises from the auto-ignition of the end gas ahead of the propagating flames, i. e. after the spark time, and develops in the opposite direction with respect to the main combustion front. It can be divided in two categories according to the intensity: light knock and heavy knock. The knock intensity (KI) is measured in bar and can be defined as the maximum amplitude of the fluctuating pressure signal computed by applying an high-pass filter on the pressure trace. This allows to remove from the signal the component associated to the mean value and observe only the component associated with the knock fluctuation. If we observe figure 2, we can see that the blue curve corresponding to 100% engine load has a peak amplitude of 26 bar, meaning that the knock intensity is $KI = 26$.

In lean-burn gas engines, as shown in figure 3, the optimum performance point is close to a knocking or misfiring situation. The reason for this is that, once the air-fuel ratio is optimized, to increase the efficiency of the combustion cycle it is necessary to increase the BMEP, and this generates favorable conditions for knocking. As a consequence, a certain level of knocking, within a permitted knock intensity, is tolerated. This accepted knock is the light knock.

Heavy knock is defined as conventional knock with a KI between 1 and 10 times the permitted value. This knock must be avoided to prevent engine damage and is the object of a part of the simulations performed for this study. Conventional knock, as a rule, occurs continuously once the conditions are adequate and keeps occurring in every cycle until an adverse change in the conditions takes place. (Wang, Liu, Song, et al. 2014)

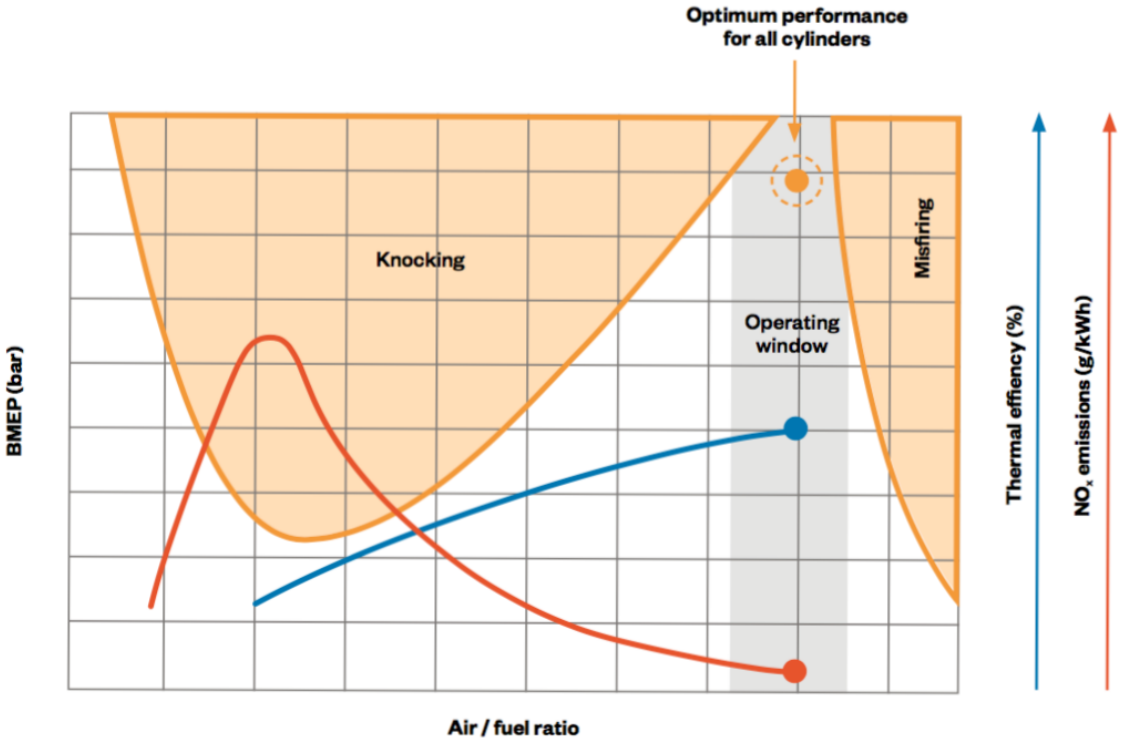

Figure 3: Optimal operating window of a lean-burn gas engine utilizing the Otto combustion cycles (Gassner et al. 2016)

In super knock the processes occurring inside the combustion chamber have been deeply investigated in the last 10 years by the scientific community and are significantly different from the ones occurring in normal knock. In general, all the authors agree that this kind of knocking is associated with pre-ignition and, beside the chemical aspects, the main difference with conventional knock is the amplitude of the pressure oscillations around the mean combustion pressure. The phenomenological aspects of heavy knock are commonly a peak pressure of 50% above the average peak pressure of normal combustion and the pressure oscillation above one order of the permitted knock intensity. Differently from conventional knock, super-knock occurs in a stochastic way and therefore it is impossible to predict when it will occur. (Wang, Liu, and D. Reitz 2017)

2.2 Cylinder overpressure

Another risk that threatens modern engines with high boost is cylinder overpressure. Cylinder overpressure can be defined as excessive pressure inside the cylinder during the compression-combustion cycle. There are several reasons for the occurrence of this phenomenon but all of them lead to the excessive admission of air inside the cylinder. Possible roots of the problem can be, for example, wrong inlet valve timing or a sensor failure, but also a mechanical failure of the valve mechanism that would leave the inlet valve open and let the turbocharge system over-boost

the engine. Figure 4 represents a comparison between a normal pressure curve at

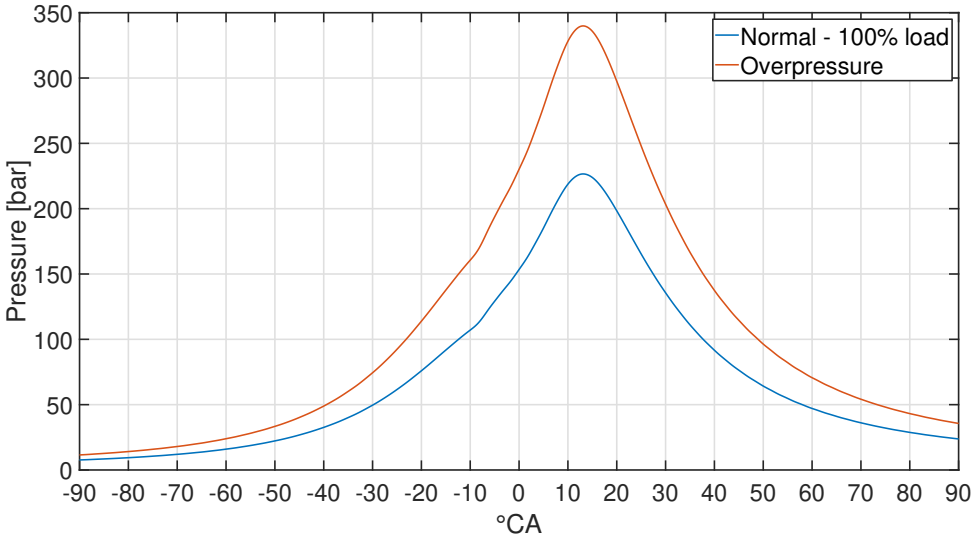

Figure 4: Comparison between a normal pressure trace at 100% load and a situation of overpressure

full load and a situation of overpressure generated by an inlet mechanism failure. The pressure is higher along the whole compression cycle because the amount of air present at the beginning of the compression is more that the design value and, when the combustion occurs, the peak reaches its maximum. The peak value is used to evaluate the intensity of the phenomenon.
The consequences of overpressure vary according to its intensity and can in the worst cases represent a threat for the persons operating in the engine room. In order of magnitude we can have: an increment of torsional vibrations in case of small overpressure, a blow-by, a reduction of fatigue life or in the worst cases a structural failure of the cylinder components (liner, head, sealing) or of the piston elements (piston, pins and connecting rod).

2.3 Engine misfire

A third phenomenon that is considered in this work is engine misfire. Engine misfire can be caused by a defect in the fuel ignition system, a lack of spark in the spark plug or by a lack of ignitable mixture a the spark plug at the time of ignition. The result of such anomalies is an abrupt reduction of the engine power and consequently an increase of the specific fuel consumption (van Basshuysen and Schäfer 2007). From the phenomenological point of view, when misfire occurs, the combustion is missing and the result is a pressure curve almost symmetrical to the top dead center location angle (figure 5). Missing the combustion, we have a compression followed by an expansion, so the amount of energy absorbed in the first phase is released during the expansion without adding anything extra to the cycle. Connected with the reduction

of power, it is associated with a distinctive torsional vibration due to the lack of combustion in the misfiring cylinder, which generates a situation of unbalance.

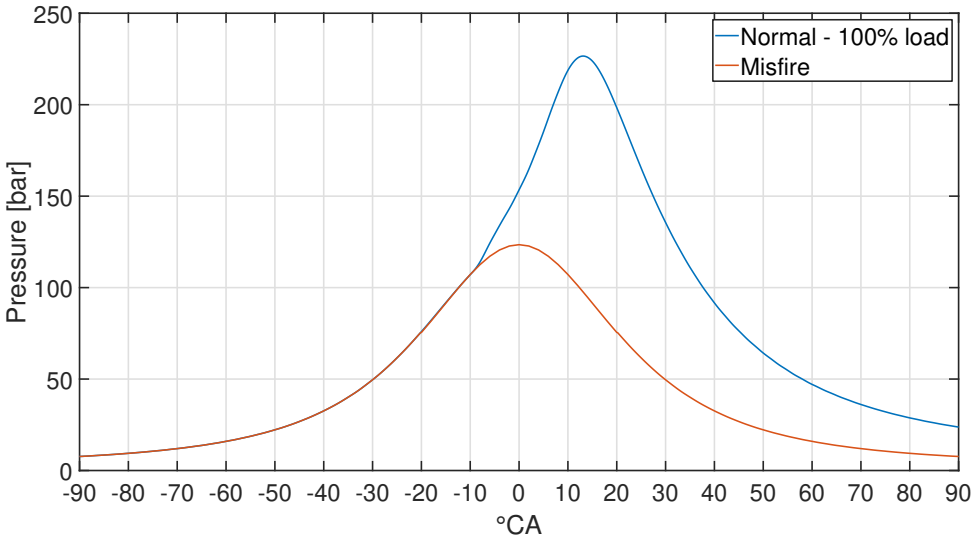

Figure 5: Comparison between a normal pressure trace at 100% load and a situation of misfire

2.4 Anomaly detection algorithms

The goal of anomaly detection is to identify unusual patterns that differ from a normal behavior. In data analytics these unusual patterns are called outliers, meaning that they are detached from what is grouped and labelled as normal. Anomaly detection can be carried out using two main approaches: statistics-based methods and machine-learning methods.

In statistics methods, the data are analyzed using statistical functions as mean, median, mode and quantiles to determine intervals or ranges within the values that can be considered normal and flag the values that are outside those ranges. They generally work in two stages: training where the statistical model is fitted, and a test stage where new values are analyzed.

In machine learning methods, the algorithms undergo first a learning phase with a training dataset where the algorithms understand by themselfes what is normal and what is not and then they can be used to analyze new values. These can be generally divided into two categories: pattern recognition algorithms (PRA) and artificial neural networks (ANN). The difference with the statistical methods is that the machine learning algorithms adjust their parameters by themselfes to fit the training dataset. (Chandola et al. 2009)

In this research work the focus is on machine learning algorithms, and it was decided to evaluate the performance of a selection of pattern recognition algorithms and a simple neural network designed for the same objective.

2.4.1 Pattern recognition algorithms

Pattern recognition algorithms are a group of algorithms devised to recognize patterns within a dataset.
It is possible to divide them in sub-groups on the basis of their way of learning; the two main groups are supervised learning and unsupervised learning. Supervised algorithms require that in the training phase the algorithm is provided with a set of samples that are labelled. The labels are used to tell the algorithm which class each and every element belongs to. In the unsupervised learning, on the other hand, the algorithm is provided with the whole training data set without any information on the category of origin of the samples. Clustering, for example, is a typology of unsupervised learning where the dataset is parted into homogeneous regions on the basis of the learning capabilities of the algorithm itself. (Mohri et al. 2012)
For the purpose of identifying the anomalies, in addition to the mere detection, the supervised learning class of algorithms is the preferred choice.
The k-nearest neighbors algorithm is a well-known and studied supervised-learning algorithm that classifies data on the basis of the information that the algorithm receives during the training on the basis of the location of the data in the domain (Altman 1992). The output of this algorithm is the class of the sample that is given as input. In a simplified way: the algorithm receives as input a sample (for example, represented by an array of numbers) and locates it in the space together with the labelled values that the algorithm has received during the training. Then it reads the labels of the k-nearest-labelled samples (the neighbors) and counts which class is the most popular. Based on that analysis, the algorithm then assigns the said class to it. A common improvement adopted in this approach is giving a weight to the class of the neighbors on the basis of their distance from the sample that has to be classified. (Zhang 2013)
The support vector machine is another popular pattern recognition algorithm. Its working principle is based on the definition, during the training phase, of an hyperplane that separates the different classes in the spatial domain. This hyperplane divides the domain in regions and each region corresponds to a known class. This approach is defined non probabilistic because the boundary between the classes is rigidly defined by the hyperplane (Mohri et al. 2012). This algorithm is particularly suitable for cases of binary classification, but it can be implemented also for multi-class classifications problems (Hsu and Lin 2002) and for probabilistic classifications (Platt 1999).

2.4.2 Artificial neural network

Artificial neural network algorithms are a family of algorithms inspired by the way of working of human neural networks that constitute the human brain.
The idea that generated the modern neural networks was suggested by Alan Turing in the late forties with a machine that was able to self-organize itself on the basis of binary choices (Ince 1992). Since then, neural networks have been deeply studied and implemented for decades and nowadays they are a part everybody's ordinary life. They are utilized to make prediction of people's behavior, voice recognition, self

driving vehicles, data mining, electricity consumption forecasts, medical diagniosis, etc. (El-Shahat 2018)
A neural network is constituted by elements, called neurons, which are arranged in layers and interconnected between themselves layer-to-layer. Each interconnection

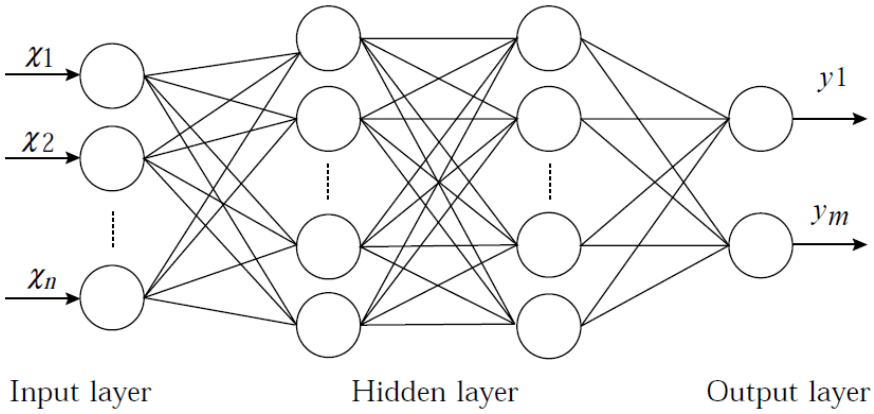

Figure 6: An example of a simple neural network architecture with two hidden layers (Yu et al. 2007)

building the network takes the output of a neuron from a layer and transmits it to the input of another neuron in the adjacent layer. In supervised-learning networks, each connection represents an equation whose parameters are tuned in the training phase in order to have the correct output in the last layer. In unsupervised-learning networks, the network itself tries to create a classification of the inputs and adjust the weights of the equations accordingly. Usually a network is composed by two external layers, called input layer and output layer, and one or more hidden layers located in between. (Yu et al. 2007)
The ANN tested in this work is a 2-layer feed-forward network with sigmoid hidden neurons and softmax output neurons. The structure of this network is shown in figure 7.
Sigmoid hidden neurons belong to a cathegory of neurons in which each neuron receives the inputs and performs a weighted sum of them, and the said sum is then processed by a transfer function that has a sigmoid shape. The result of this function

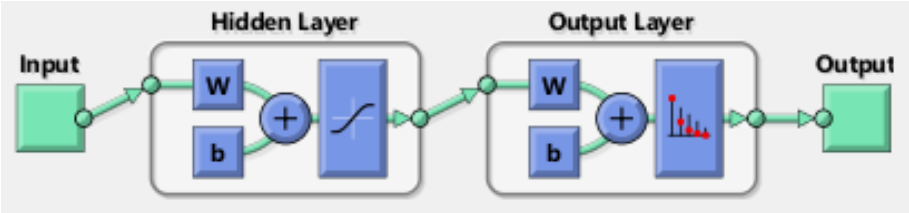

Figure 7: Structure of the neural network utilized (Beale et al. 2018a)

is the output of the neuron. The benefit of a sigmoid shape is that it is a non-linear function with an easy to calculate derivative, this aspect reduces the computational

load in the training phase because the parameters of that functions are among the parameters that are adjusted within the network during the training phase. (Jun and Moraga 1995)
The softmax neurons, similarly to the sigmoid neurons, perform a sum and then apply a function. In this case the function is a softmax transformation and the aim is to categorize the output on a probabilistic basis. (Bishop 2006)

3 Methods

3.1 Torsional vibration model

The engine generating set object of this study is simulated and analyzed using the software GT-SUITE by Gamma Technologies. GT-SUITE is a 1-D visual-modeling simulation tool that allows to create models in a user-friendly graphical programming environment. Each block in the model represents either a component or a function and hides behind its interface a set of mathematical equations and boundary conditions that allow the interaction with the other blocks present in the model.
To evaluate the torsional vibrations in the engine, the main components of the engine, a generator and a coupling are included in the simulation. The model is then tuned and validated using real data measured from an identical real engine generating set. In figure 8, visible on a bigger scale in figure 9, there is a representation of

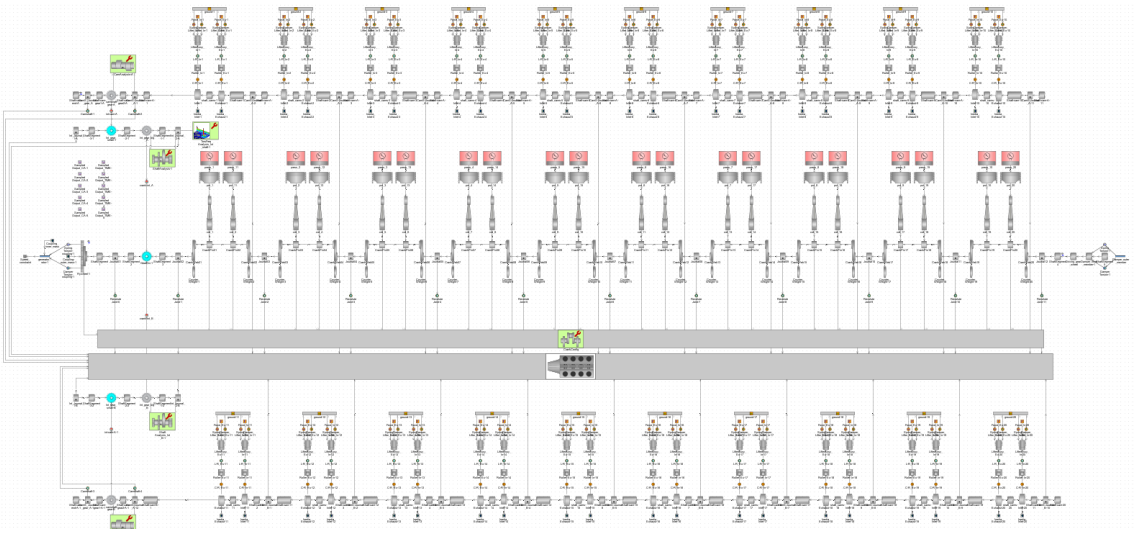

Figure 8: The view of the full model as is appears in the GT GUI

the model in the GT GUI. From the picture, it is possible to see how the different components are connected together by means of logical links (lines or arrows) and how it is possible to sort the components in a clear visual way.

3.1.1 Engine

The engine is a V20 gas engine. Each bank has a single camshaft connected to the flywheel through an intermediate gear shaft that also reduces the speed of the camshaft to half the speed of the crankshaft. At the ends of the crankshaft, there is a torsional vibration damper on one side and a flywheel on the other.

Crankshaft The crankshaft is a component with an extremely complex geometry. Its assembly includes ten throws, twenty counterweights, several geometrical features as chamfers and fillets, lubrication ducts, connection flanges, and so on. In the

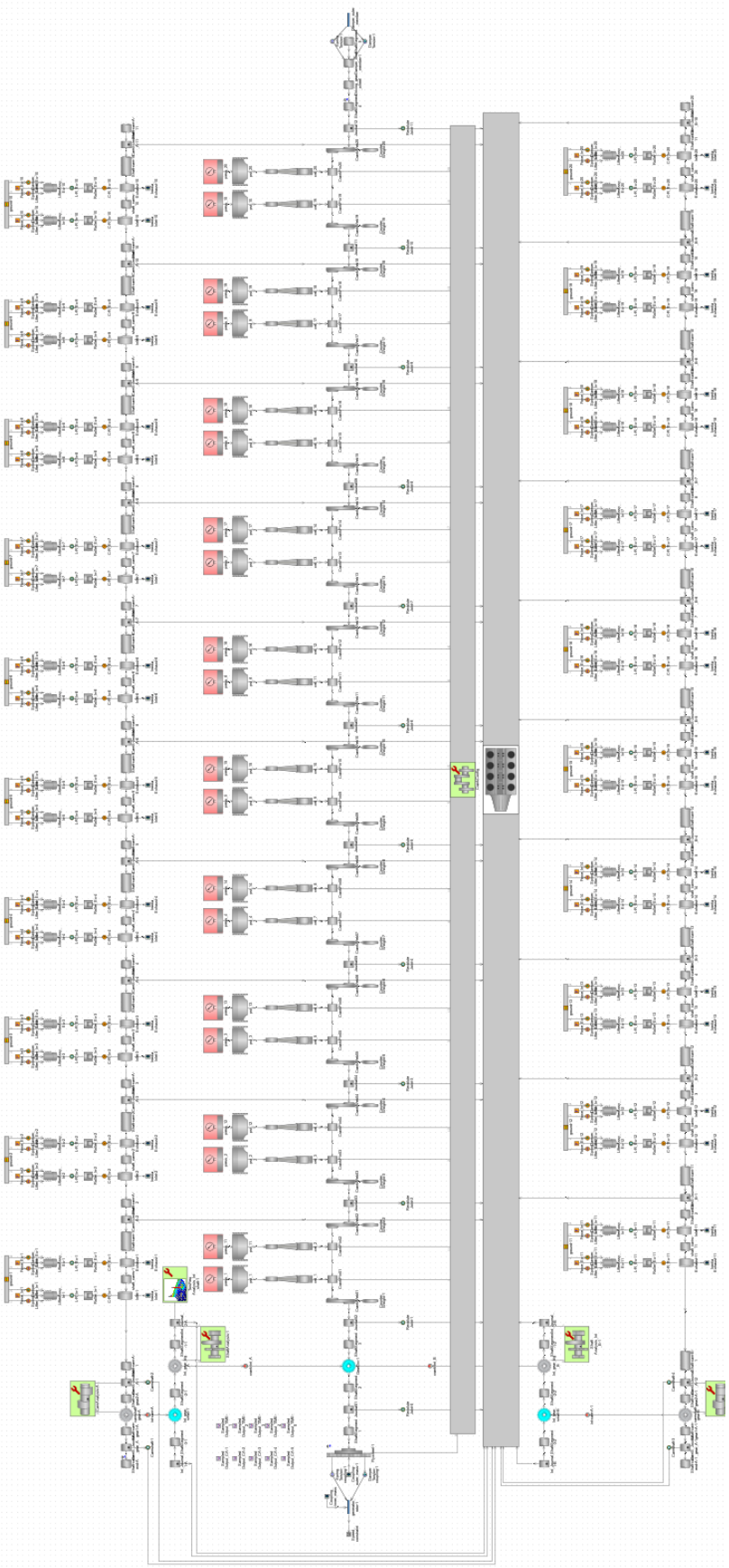

Figure 9: The view of the full model as is appears in the GT GUI

simulation, the crankshaft is discretized in its main constituent features: bearings, crank-webs, journals, shaft segments, counterweights, and gears. Every discrete component is characterized by its own set of equations and parameters and the components are connected by nodes as in the classical FEM approach. In this way, it is possible to achieve a high level of accuracy in the description of the behavior of the crankshaft even if the number of nodes is reasonably low. Also, dividing the shaft in discrete parts allows to query the calculated values node by node.
To have an accurate reading of the dimensions the geometry has been first imported from the CAD model (figure 10) of the crankshaft assembly into the software GT-SPACECLAIM and then divided in its constituent sub-components with the GT tool GEM3D. Figure 11 shows the crankshaft after the subdivision: the journals are in red, in green the crank webs, in brown the left bank crank pins, in bronze the right bank pins, in light blue the counterweights, in blue the shaft segments, and in grey the gear to which the intermediate shafts are connected. From the sub-components,

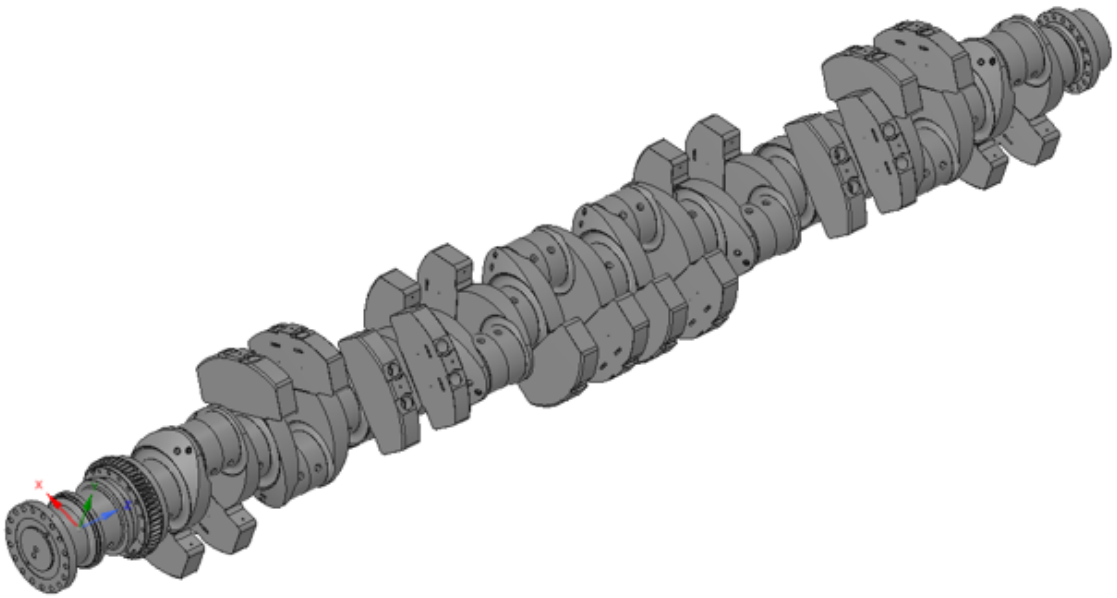

Figure 10: CAD model of the crankshaft assembly

GT is able to translate the geometry into geometrical parameters. In such a way, the features of the original model are accounted for and the elements composing the crankshaft have an accurate calculation of the equivalent properties (e.g. stiffness and damping).
In the simulation, nevertheless, some simplifications were needed:

- Lubrication holes and ducts, bolt holes, chamfers and fillets are not taken into account.
- The journal axes are constrained along the axis of rotation.
- The pin axes are constrained to the connecting rod big end axes.

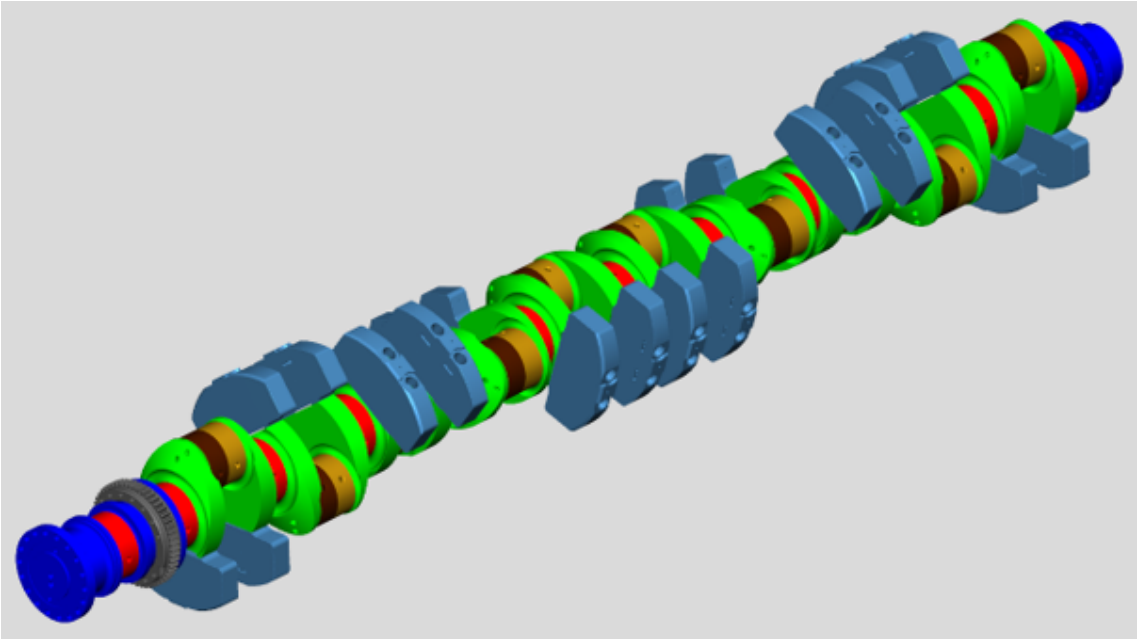

Figure 11: Discretization of the 3D model

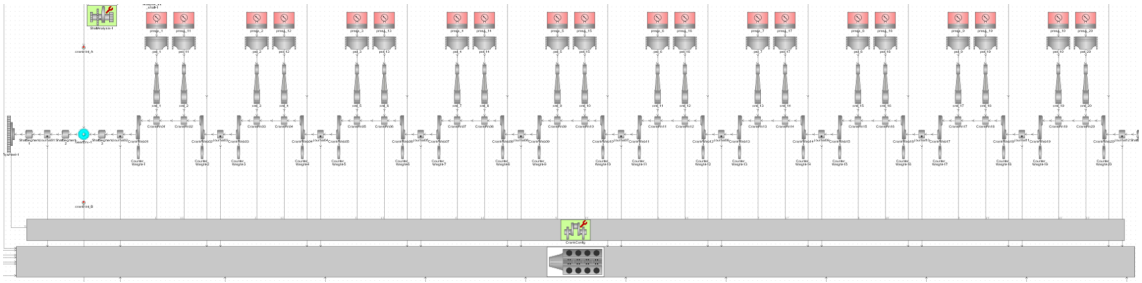

Figure 12: GT model of the crankshaft

- The crankshaft is studied only from the torsional point of view, thus bending is not considered.

Vibration damper A torsional vibration damper is located at the free-end of the crankshaft. The main function of this component is to modify the natural frequency of the crankshaft to eliminate most of the critical resonance. In addition, given the fact that in the same location several auxiliary systems are connected, this damper covers the double function of reducing the torsional vibrations in the crankshaft and providing a more stationary power output for those systems. The auxiliary devices, working at higher frequencies than the crankshaft, generate excitations that can be neglected.
The damper is composed by two masses interconnected by an oil-spring system, where the oil is provided under pressure from the crankshaft lubrication system. In the GT model, this component is simulated with two shaft segments with equivalent polar moment of inertia which are connected one to the crankshaft via a rigid connection

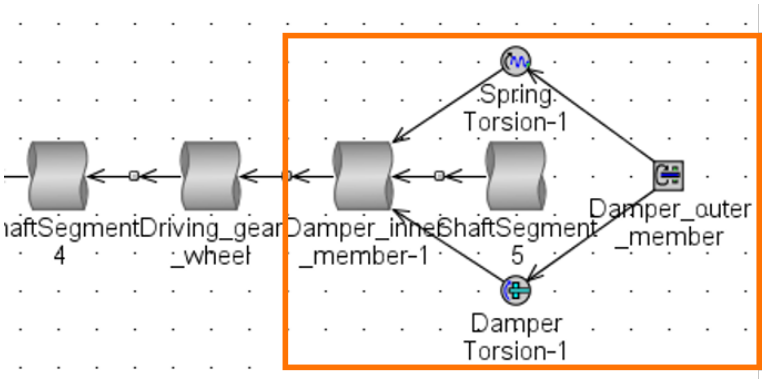

Figure 13: Damper portion of the GT model

and the other to the first shaft by mean of a torsional spring and a torsional damper. The said spring and damper are adjusted to match the real damper specifications.

Flywheel The flywheel is used to provide angular momentum to the rotating component of the engine and as a connection point for the coupling to the generator. It is connected to the front end of the crankshaft with a rigid connection and is the point from which the speed is logged, both in the simulation and in the real engines. In the real medium bore production engines, the acquisition is performed with an inductive sensor acting on a toothed/slotted path machined on the inner side of the flywheel with 120 slots over the whole circumference. The resolution of this system is 3^o and the accuracy 0.04^o (Storm et al. 2017). This, together with the analog-to-digital sampling frequency of 2000 Hz, provides a speed signal which is reliable up to 200 Hz.

Gear train The crankshaft is connected to the camshafts via two intermediate shafts that act also as reduction gears. These shafts represent a substantial source of rotational vibrations on the system for two reasons: first, because of the uneven load generated by the camshaft (noise at the source) and, secondly, for the small impacts between the teeth in the double straight-cut gear connections camshaft-to-intermediate shaft and intermediate shaft-to-crankshaft (noise in the transmission). GT allows a precise calculation of the gear links. To calculate the gear mesh, it takes into account the normal nominal backlash, the contact ratio and the contact stiffness and damping, together with the individual gear design specifications; the data used come from measurements taken directly from the test engine. On the other hand, the bending of the whole intermediate shaft is less relevant, thus also here it is considered only from the torsional point of view.

Camshafts The model of the camshafts is the most complex part of the GT model. The benefit of including also the camshafts in this model is to account for the noise generated by the opening and closing of the engine valves.
The engine object of this simulation has a single camshaft per bank, and the cams are acting through a roller-tappet element on a hydraulic mechanism that allows

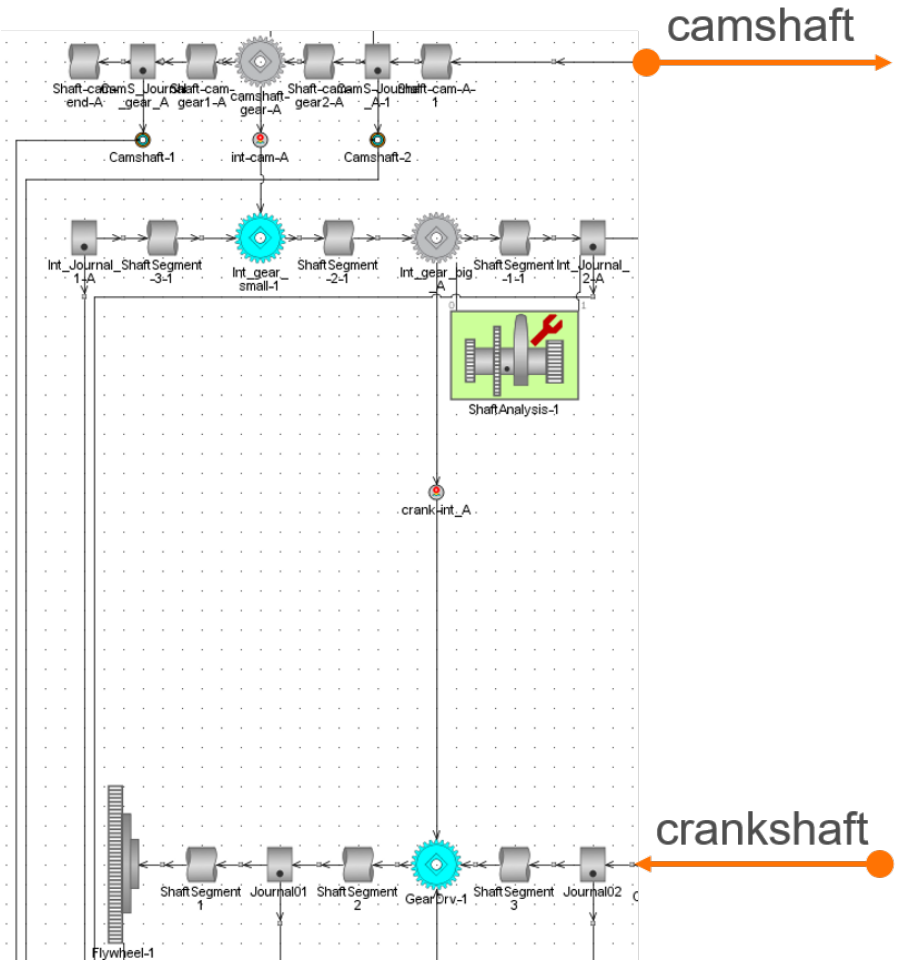

Figure 14: Cranktrain portion of the GT model

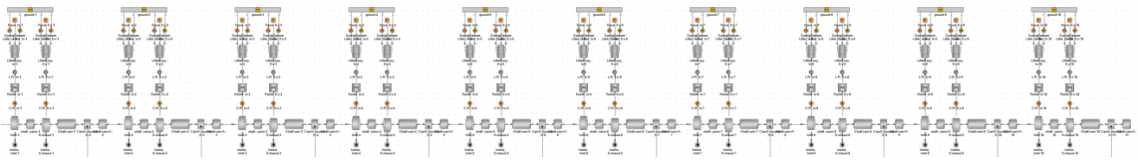

Figure 15: GT model of the left camshaft

to control the valve closing timings depending on the engine load (Sundsten 2013). This hydraulic mechanism is not entirely simulated, as visible on figure 16 where there is a detail view of the camtrain of the first cylinder. A force, a spring element, and a damper element are acting on the lifters. The force elements correspond to the pressure generated in the hydraulic piston of the cam mechanism and have been calculated separately to lighten the computational load of the simulation. The profiles of these forces, one for the intake and one for the exhaust, are visible in figure 17 as a function of the crankshaft angle. The spring and damper elements are adjusted to guarantee a constant contact between the lobes and the rollers in the portion of the cycle where there is no pressure in the hydraulic system because the

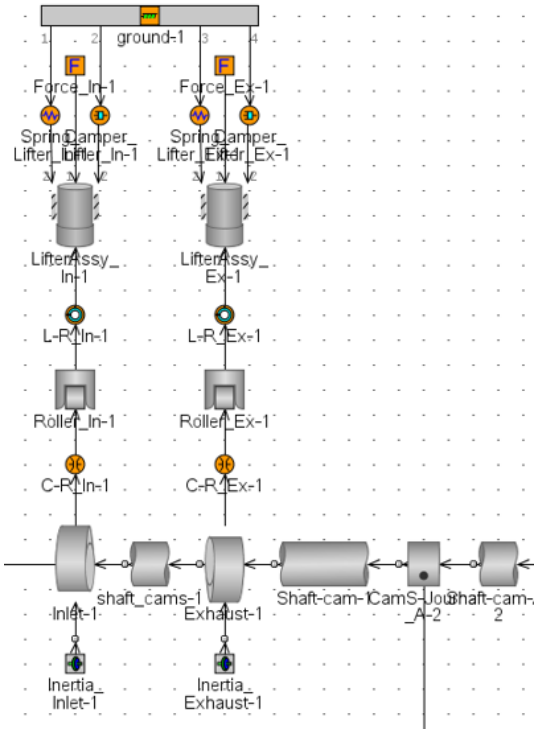

Figure 16: Detail of a camtrain

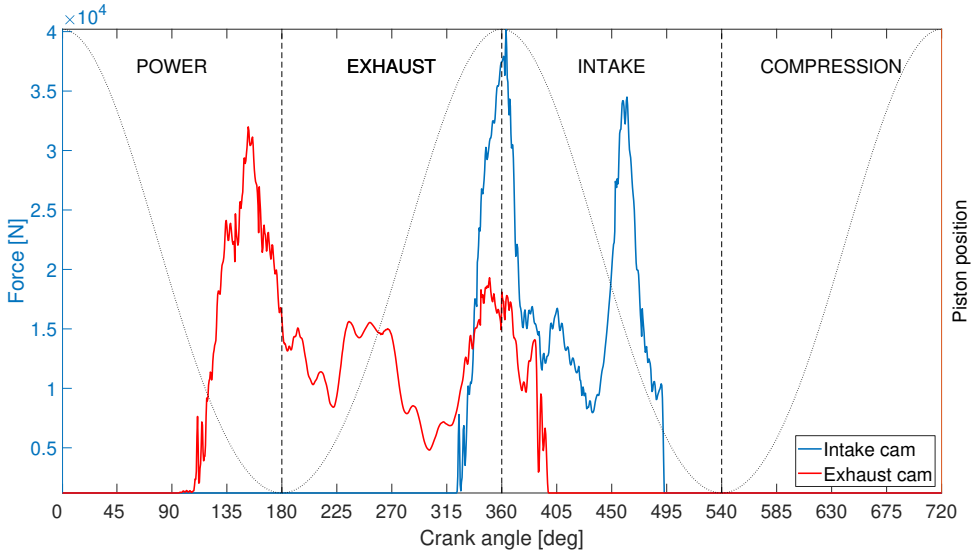

Figure 17: Forces acting on the camshaft lobes

valves are closed. In those portions, the valves are kept closed by pre-loaded springs as in the traditional camtrain systems.
GT is really accurate in calculating the contact between elements (Mohiuddin et al. 2008). In this case the simulation of the contact between the roller and lobe takes into account the damping generated by the lubrication film present on the lobe and

the material stiffness for determining the contact surface, applying the Hertz theory on the contact between two elastic bodies (Hertz 1881).
Similarly to the other shafts, also the camshafts are analyzed only from the torsional point of view, constraining the radial movement of the shaft in correspondence with their journals.

3.1.2 Coupling

The coupling is a flexible element used to transmit power from the engine to the generator. The reason behind the use of this element is to account for the small

Figure 18: Illustration of a real coupling (courtesy of Renold)

misalignments that are present due to manufacturing tolerances and the assembly of the components. A secondary, but not less important, function of the coupling is to

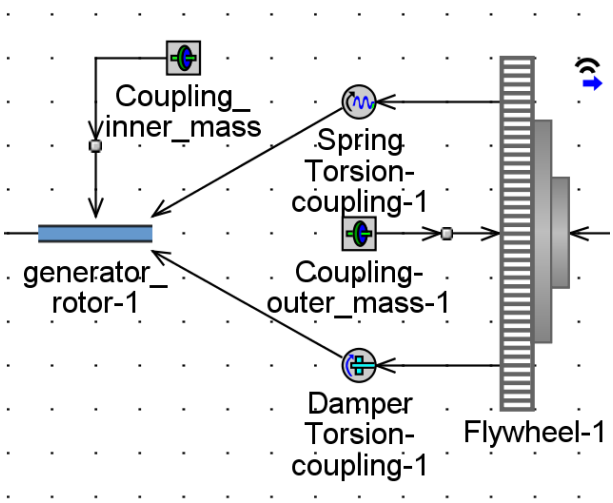

Figure 19: Coupling portion of the GT model

reduce the transmission of the torsional vibrations generated by the engine to the generator.
The coupling in use is composed of two elements that resemble an internal gear and

an external gear, with a small number of teeth, that are matched together (figure 18). In the space between the teeth, rubber elements are located in order to avoid the direct contact between stiff parts and to allow the misalignment between the two sides.
In the GT model (figure 19), the coupling has been modeled taking into account the inertia of the system and the power transmission.
For the first aspect, two rotational inertia elements attached to the flywheel and to the generator rotor respectively have been set up to match the polar moment of inertia of the coupling elements.
Regarding the power transmission aspect, a torsional spring and a torsional damper are used to connect the generator shaft to the flywheel. The characteristic values of these last two elements, stiffness and damping coefficient, have been adjusted to include not only the properties of the coupling itself but also the contributions of all the non-modeled elements. This aspect will be analyzed more in detail in the section 3.2 regarding the validation of the model. The non-linear behavior of the rubber elements can be neglected in this case because the model runs in steady-state conditions, the properties of the material have been provided by the manufacturer for the operative conditions (load and temperature).

3.1.3 Generator

The generator used in the generating set is a prototype developed by ABB and is connected to the Spanish national electricity grid. The speed of the generator is bond to the frequency of the grid. This means that, depending on the load connected to it, it can either generate or absorb current but the speed will not change unless the frequency of the grid changes.
Such precondition allows a strong simplification when building the torsional vibration model. As visible in figure 20, the rotor is simply modeled as a shaft with equivalent

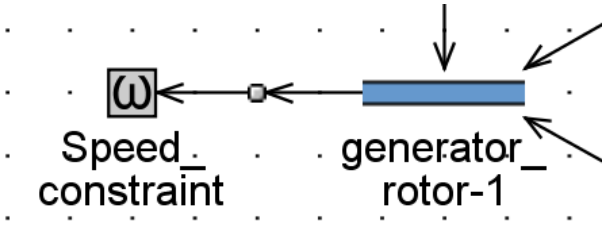

Figure 20: Generator portion of the GT model

inertia where attached to it there is a speed constraint that locks the mean speed to around 750 RPM. This approach allows anyway enough freedom for the flywheel to fluctuate as in the real set-up due to the shaft elasticity and the presence of the coupling, and at the same time it guarantees that the engine will operate at the expected speed throughout the whole simulation time.

3.1.4 Assumptions

This kind of dynamic simulations follows the approach defined by the finite element method and can be reduced to a set of governing differential equation of motion of the kind of

$$\dot{\underline{y}} = \underline{f}(t, \underline{y}), \tag{3}$$

where t denotes time, $\underline{y}$ is a vector that contains the unknowns of the system, $\underline{f}$ is a non-linear function of t and $\underline{y}$, and the dot operator denotes the derivative with respect to time.
GT-SUITE offers a wide variety of solvers to calculate the solution of 3. In general they can be grouped in two main families: implicit integrators and explicit integrators. Explicit integrations require a small time step to maintain the stability of the solution. For this reason they are well suited for non stiff problems where high frequencies are not involved.
Implicit integrations are better suited for stiff problems (Hairer and Wanner 1996). The differential equations of motion generated from a torsional simulation model as the one of this research project are stiff, nonlinear, differential algebraic equations, and the stiffness generates high frequency excitation in the elastic mechanic components that, with an explicit integrator, would require a considerably small integration step. The integrator chosen for this problem is thus the Spectral Deferred Correction (S.D.C.) - 3 stage. It is an implicit integration system derived from the 3-stage Radau IIA algorithm accurate up to the 5^{th} order, and it possesses excellent stability properties and numerical damping.
The incremental stage variables $\underline{z}_i$ can be expressed as

$$\underline{z}_i = \hat{\underline{z}}_i + \underline{\varepsilon}_i \ for \ i = 1, 2, 3 \tag{4}$$

where $\hat{\underline{z}}_i$ are approximations to $\underline{z}_i$ and $\underline{\varepsilon}_i$ are the truncation errors.
The form of the integration scheme can be described in the following way:

$$\begin{pmatrix} \underline{f}(t_n + c_1 h,\ \underline{y}_n + \hat{\underline{z}}_1 + \underline{\varepsilon}_1) \\ \underline{f}(t_n + c_2 h,\ \underline{y}_n + \hat{\underline{z}}_2 + \underline{\varepsilon}_2) \\ \underline{f}(t_n + c_3 h,\ \underline{y}_n + \hat{\underline{z}}_3 + \underline{\varepsilon}_3) \end{pmatrix} = \frac{1}{h} \begin{bmatrix} a_{11}I & a_{12}I & a_{13}I \\ a_{21}I & a_{22}I & a_{23}I \\ a_{31}I & a_{32}I & a_{33}I \end{bmatrix} \begin{pmatrix} \hat{\underline{z}}_1 + \underline{\varepsilon}_1 \\ \hat{\underline{z}}_2 + \underline{\varepsilon}_2 \\ \hat{\underline{z}}_3 + \underline{\varepsilon}_3 \end{pmatrix} \tag{5}$$

where t_n is the time, h is the step size, defined as $h = t_{n+1} - t_n$, a_{ij}, and c_i are stage parameters derived from the Radau IIA method (Hairer and Wanner 1996).
The calculation of the $n + 1$ solution is obtained by solving the equations 5 for the numerical truncation error and updating the approximate incremental variables $\hat{\underline{z}}_i$ with $\hat{\underline{z}}_i = \hat{\underline{z}}_i + \hat{\underline{\varepsilon}}_i^k$. Then the solution takes the form

$$\underline{y}_{n+1} = \underline{y}_n + \hat{\underline{z}}_3 + \underline{\varepsilon}_3 \tag{6}$$

3.2 Model Validation

The model validation takes place in two stages: first a comparison between the spectra of the flywheel speed of the real engine and the model, then the analysis

of the behavior of the engine when simulating an anomaly, looking for an expected variation. The anomaly chosen for the validation, the misfire, has an effect on the spectrum of the speed which has been studied in literature (Cavina et al. 2002) and that is known, thus it is suitable for this purpose. The other anomalies, overpressure and heavy knock, are simulated assuming that the model, once validated, is capable of reacting accordingly with the new pressure curves.

3.2.1 Flywheel speed analysis

Measured pressure traces The input adopted for the validation of the model is the entire set of pressure traces measured from the real engine prototype. The engine is running at nominal speed (750 RPM) and 96% of the maximum design load. The measurement starts once a steady state is reached. The pressure is sampled using a Dewesoft measurements apparatus at the frequency of 20 kHz by 20 analog sensors with 30 bar/V resolution.
The data set is stored in a time domain. Hence, to be able to feed it to the GT model, a conversion from a time domain to a crank angle domain is necessary. Therefore, it is necessary to assume the rotational speed of the crankshaft constant so that there is a direct connection between time steps and crank steps. The increment of CA in degrees per second is

$$750\ RPM \rightarrow 12.5\ RPS \rightarrow 4500\ deg/s, \tag{7}$$

and the increments in the two domains, obtained by dividing the increment per second by the sampling frequency, are:

$$\Delta s = 0.00005\ seconds \tag{8}$$

$$\Delta CA = 0.225\ CA. \tag{9}$$

The gas combustion cycles have peak pressure values that, in this class of engines, may vary by ±25 bar around the cycle-to-cycle average of peak values (Sen et al. 2010). For this reason, to be able to compare simulation and real data, it is fundamental to use all the pressure traces measured in the corresponding cylinders and with an exact alignment.
Since the measurement starts from an unknown point, it is necessary to align the measured profiles with a known one in order to link the pressure values to the correct corresponding crank angle values. In figure 21, the measurement of the pressure in cylinder one is compared with the reference curve at 100% load.
The compression phase is not affected by the load and thus can be utilized as reference for the alignment. Close to the peak, the difference of load is visible, and the reference curve continues to climb for a few more degrees due to the greater amount of fuel present in the mixture. From that point onward, for the whole expansion phase, the design curve shows higher pressure values.

Speed comparison In the test engine, the speed is recorded using a Baumer linear encoder and a magnetic belt located on the crankshaft shaft segment between

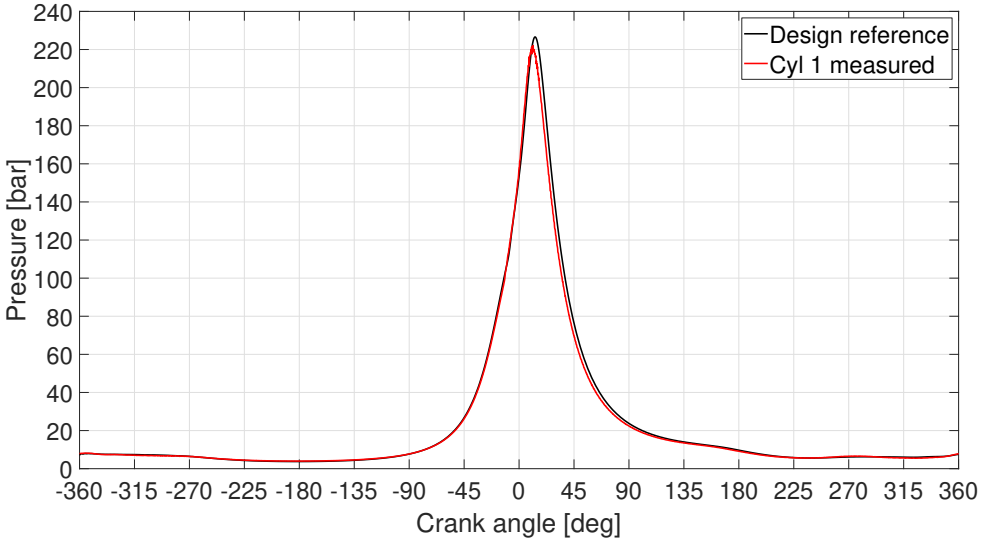

Figure 21: Comparison between design value and measured data

the flywheel and the engine block. The magnetic belt generates 2048 pulses per revolution wich are converted from the analog sine-cosine signal to a pulse digital signal (Nyce 2004; Walcher 1994) with a sampling frequency of 50 kHz.
The speed signal is compared both in the time domain (figure 22) and in the frequency domain (figure 23). Figure 22 shows in the time domain the values of the flywheel speed over two combustion cycles (marked with yellow dashed lines). There, it

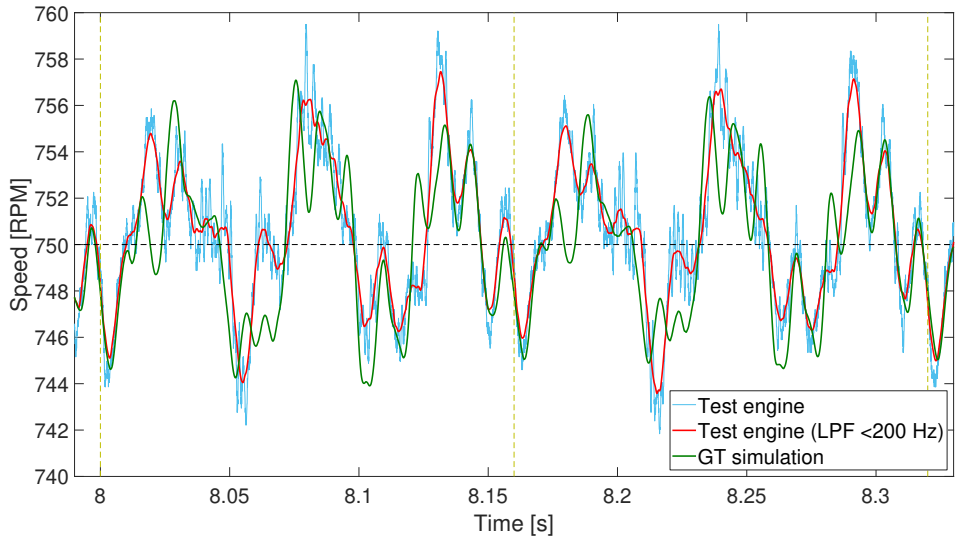

Figure 22: Comparison between measured and simulated flywheel speed

is possible to see a strong correlation between the speed signal of the test engine (blue line) and the GT simulation (orange line). To increase the readability of the graph, the speed signal is also filtered with a low-pass filter (LPF) to remove the

effects of high frequency noise (red line). This noise is principally generated by auxiliary systems, such as high speed pumps, which are not modeled in this torsional vibration model. For this reason, it is evident that the filtered signal shows a higher correspondence with the simulation in comparison with the raw signal.
In addition to the time domain analysis, in figure 23, the spectra of the raw speed signal and the simulation result are put next to each other. The most important

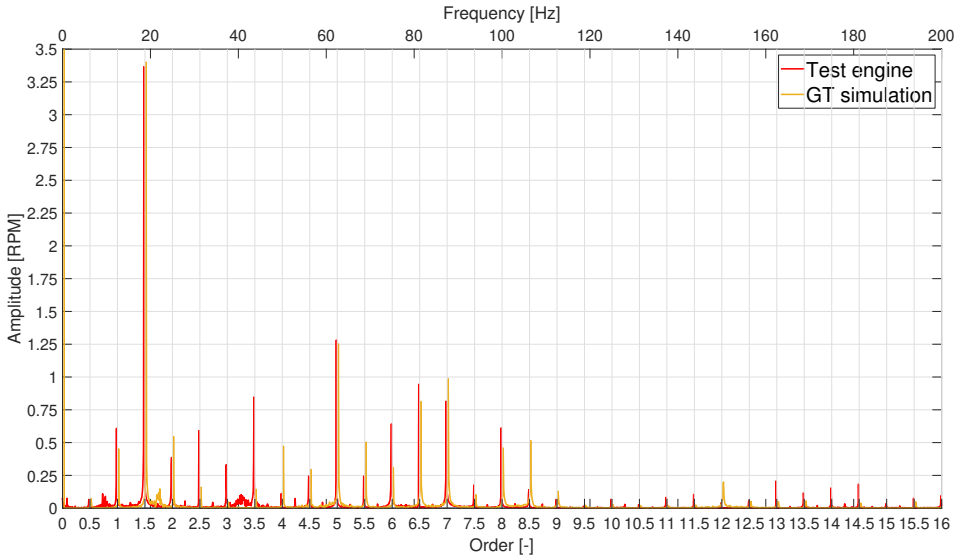

Figure 23: Spectral comparison between measured and simulated flywheel speed. The plot shows the constituent waves of the speed signals (in RPM) at their corresponding frequencies. The value at 0 Hz, which is the average speed, is off the scale.

peaks are the ones corresponding to orders 0.5, 1, 1.5 and 5 because those are directly connected to the firing frequency. From that figure, it is notable that almost all the peaks of the spectrum of the test engine find a match with the peaks of the GT model. The only peaks that are not matching are the ones corresponding to orders 2.5, 3 and 3.5 due to a resonance present in the test engine but not appearing in the GT model where the peaks are considerably lower.

3.2.2 Expected behavior analysis in presence of an anomaly

Engine misfire In case of one cylinder misfiring the engine is unbalanced because not all the cylinders have the same power output. This is visible in the spectrum of the flywheel speed, when compared with the spectrum of a normally working engine, because the orders listed in the previous paragraph are affected.
To simulate a misfiring situation the pressure trace of cylinder one is replaced repeatedly in all the cycles with the one illustrated in figure 5. In such a way it is possible to reach a steady condition and calculate the spectrum over the whole simulation time. In this way, the small variations that would be visible in the spectra, if calculated over a period of only one or two engine cycles, are averaged and it is possible to do a more robust comparison. Those variations are due to the

non-uniformity in peak pressures from cycle to cycle that were discussed before and that are present in the normally-working cylinders.

Spectral analysis In figure 24, the spectrum of the simulation in normal conditions is compared with the spectrum of the simulation with misfire simulated in cylinder 1, and the peaks of interest are highlighted with arrows. It is visible that all the peaks that are directly involved with the combustion frequency are subjected to a change in amplitude. This proves that the model is behaving correctly.

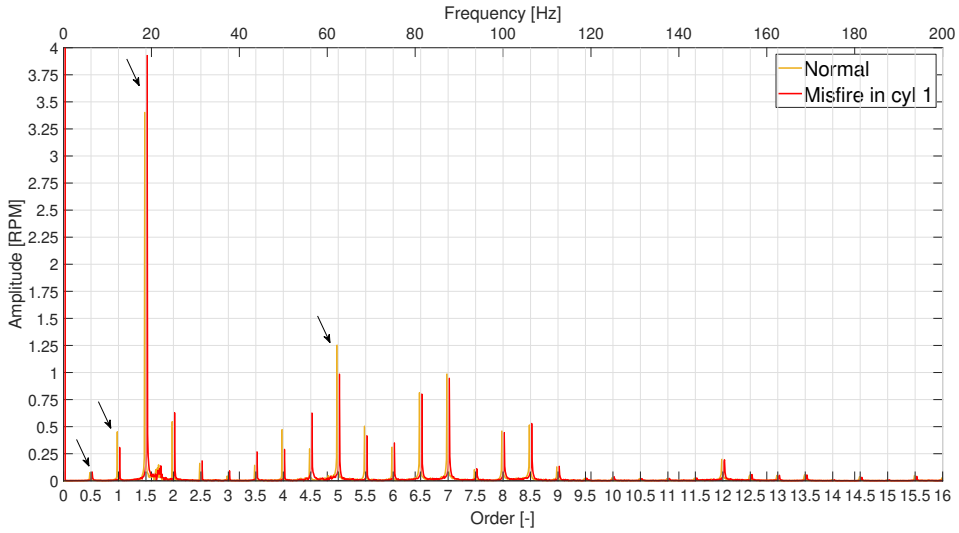

Figure 24: Spectral comparison between normal condition and misfire in cylinder 1

3.3 Sensitivity of the model

In the tuning phase of the creation of the model, the coupling characteristic parameters (torsional stiffness and torsional damping coefficient) were used to adjust the behavior of the simulated engine to match the behavior of the test engine. This phase is not discussed in these pages due to the scarce attraction of an iterative process that, step by step, tunes those parameters until a satisfactory level of accuracy is reached. Worth citing from that process is the effect that the coupling parameters have on the engine behavior, both the inertia and the spring-damper system. It is fundamental for the correctness of the simulation output that those values are set properly in order to avoid unexpected resonance phenomena that are not present in the real scenario.

3.4 Simulations

The torsional vibration model is used to simulate several cases of normal behavior and anomalies. In total, 280 cases are simulated, and each case covers a period of 60 engine cycles. Of these cycles, the first 10 are discarded to be sure that a steady

condition is reached and the following 50 cycles are stored. More precisely, the cases simulated are:

- 100 cases of normal behavior. These cases are obtained with the permutation of the 20 available pressure traces in the cylinders of the model. Since the pressure traces have been recorded simultaneously in normal and identical working conditions, it is possible to change the cylinder in which they are used. In this way, other similar working conditions with small and within-the-tolerance differences are generated. The number of possible permutations can be calculated as

$$n_{p.\ traces}! = 20! = 2.4329 \cdot 10^{18}\ permutations \tag{10}$$

- 60 cases of heavy knocking. They are obtained by using the pressure trace with a heavy knock corresponding to propane mix and 100% load visible in figure 1 (blue line) in one cylinder and by permuting the normal traces in the remaining 19 cylinders. The faulty cylinder changes in every case so that the same cylinder is faulted in no more than 3 cases.

$$(n_{p.\ traces} + 1)! = 21! = 5.1091 \cdot 10^{19}\ permutations \tag{11}$$

- 60 cases of misfire. Obtained similarly to the cases of heavy knocking. The pressure trace showing misfire is visible in figure 5.

- 60 cases of overpressure. As for the previous two scenarios, the pressure trace used is shown in figure 4.

The objective of these simulations is to analyze the results in order to see whether the anomalies are easily recognizable.

Data processing

The output of the simulations is in the form of an ASCII text file and contains the time steps, the raw rotational speed values of the flywheel and other unused signals. A list of the properties of the simulation output is available in table 1.

Table 1: Properties of the simulation output

Encoding:	ASCII				
Precision:	double				
Sampling rate:	20000 Hz				
Signals stored:	FW speed	FW acc.	time	cum. crank angle	cycle angle
Units:	RPM	rad/s^2	s	deg	deg

The output of the simulations needs to be processed in order to be utilizable by the machine learning algorithms. The aim is to generate a table where the samples are

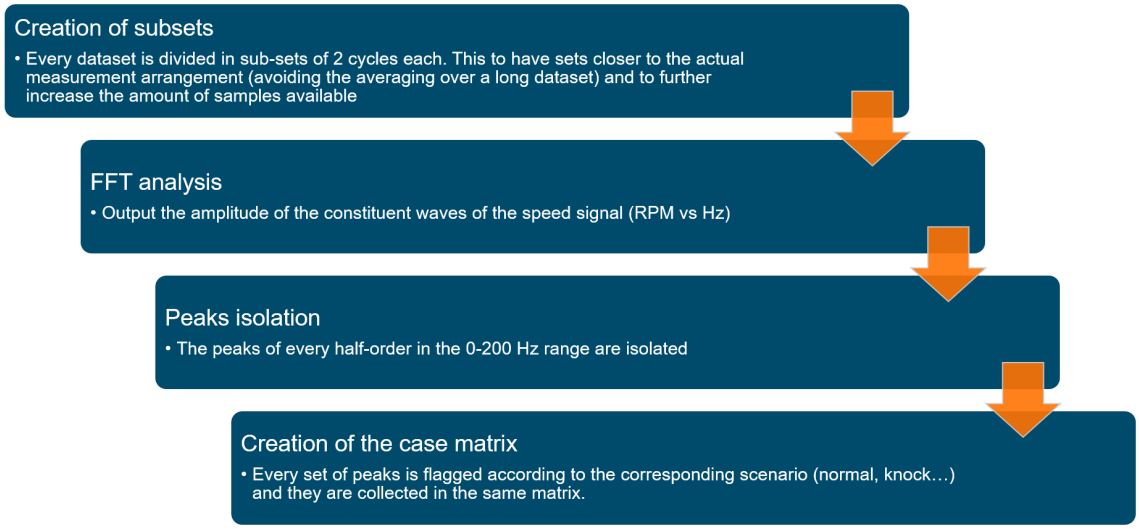

Figure 25: Workflow adopted for processing the simulation output

listed row-by-row and classified with a numeric value stored in the last column of the table itself. The work flow of the data processing is outlined in figure 25, and it is composed by 4 sequential steps:

1. First, each simulation case containing 50 engine cycles is divided in subsets of 2 cycles each. This allows to have observation periods that are closer to the actual measurement approaches adopted in engine monitoring. A secondary but useful consequence of subdividing the original set is to increase the amount of samples that will be fed to the algorithms by a factor of 25.

2. Subsequently, a fast Fourier transform (FFT) is performed on the speed signals to extract the amplitudes, in RPM, of the constituent waves. The FFT decomposes the original signal into sinusoidal oscillations at distinct frequencies. Thus, the units of the output are the original unit and Hz. The value at 0 Hz corresponds to the average speed, the values at the other frequencies are the oscillations around the average value. If all the sinusoids were summed together in the right phase the result would be the original signal.

3. The next step has the goal of isolating a frequency range of interest and cleaning the arrays of amplitudes from the non-relevant peaks. The spectra computed in the previous step are composed by a set of equally spaced peaks, and the portions in between the peaks have values close to zero. These peaks are called orders and their corresponding frequencies are multiples of the natural frequency of the system. The first peak has a frequency corresponding to the

frequency of the excitation, in this case the combustion frequency, calculated as

$$rev/min = 750\ RPM \tag{12}$$
$$combustions/min = 375\ c/min \tag{13}$$
$$combustions/s = 6.25\ c/s \tag{14}$$
$$combustionfrequency = 6.25\ Hz \tag{15}$$

The frequency range selected in this step is 0-200 Hz, and only the amplitudes corresponding to frequencies that are multiples of the combustion frequency are stored together with the speed mean value (which is actually the amplitude corresponding to 0 Hz).

4. In the last step the data is rearranged in a case matrix for the pattern recognition algorithm and in a case matrix plus output matrix for the artificial neural network. In the case matrix for the PRAs, the values of the peaks are arranged by rows, with the columns filled with the values of the peaks in ascending order, starting from 0 Hz, and the corresponding scenario label in the last column (0 for normal scenario, 1 for heavy knock, 2 for misfire, 3 for overpressure). The end result is a 7000 rows by 33 columns matrix where each row contains the peaks of the spectra of all the subsets. In the case matrix for the ANN, the matrix contains only the peak values (in the same fashion as for the PRAs) and the output matrix is a 4 columns by 7000 rows table where each column correspond to a scenario and the rows, linked by the row-number to the case matrix, contain a 1 in the cell of the corresponding case.

3.5 Testing method

3.5.1 Pattern recognition algorithms

The case matrix for the PRAs is processed using the Classification Learner application available in the Statistics and Machine Learning Toolbox 11.3 of MATLAB.
This tool allows to test a multitude of different pattern recognition algorithms and with different settings as:

- Decision tree (multiple tree sizes)
- Discriminant analysis (linear and quadratic)
- Logistic Regression
- Support Vector Machines (linear, quadratic, cubic, Gaussian)
- Nearest Neighbor Classifiers (different amounts of nearest neighbors and methods of selection of the neighbors)
- Ensemble classifiers.

The toolbox allows to evaluate the accuracy of each algorithm and configuration on the basis of a test performed with a part of the sample dataset after a training phase that takes place using all the samples available except the ones kept for the evaluation.

3.5.2 Neural network

To test the artificial neural network, I used the Neural Net Pattern Recognition application part of the Neural Network Toolbox 11.1 of MATLAB r2018a. The specifications of the neural network implemented in this app are described in detail in paragraph 2.4.2.
The app divides the rows contained in the samples matrix in three groups: training, validation and testing. The training portion, which is the biggest, is used in the learning phase where the network adjusts itself on the basis of its error in the output. The validation part is to measure the network generalization (i. e. the ability to

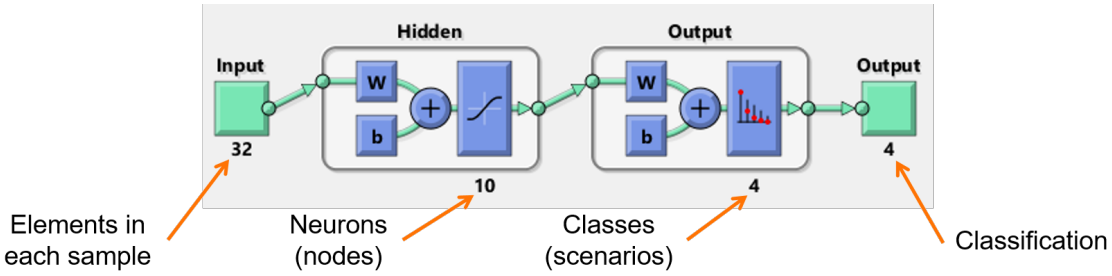

Figure 26: Artificial neural network used with key numbers

classify data that is never before seen in the classes to which the learning data belongs) and to cease the training when the generalization stops showing improvements. The testing does not affect the previos phases, but it serves to measure the performance of the network during and after training. (Beale et al. 2018b)
In figure 26, the key numbers involved in the network are listed. Every sample contains 32 values which are processed by 10 neurons and give as output 4 possible classes.

3.6 Limitations

3.6.1 Simulation model

As any simulation, results cannot represent the reality with absolute accuracy. The simulation output is affected by errors originated by simplifications done to the model and by the solver due to the finite amount of steps involved in the solution.
The torsional vibration model implemented in this simulation contains all the main components that contribute to the torsional vibrations of the engine, nevertheless smaller components that are not simulated may affect the output, especially if they are operating at frequencies close to the natural frequency of the system. One example of these components are the auxiliary pumps located at the free end of the

crankshaft which run at much higher speed compared to the crankshaft itself.
A second limitation of this kind of simulations is the lack of the operative environment in which the engine is located. To bring an example for clarity, the performance of these engines is affected by a multitude of auxiliary systems that are not modeled and always considered to work in a consistent way. In reality this does not happen and their performance varies with time.
The simulation is built with flexibility in mind. Every cylinder can be set with any pressure curve, and the only requisite is that the angle reference of the pressure trace is aligned with the TDC of cylinder number one. Hence, it is possible to simulate any kind of scenario or even multiple scenarios at the same time. The actual setup allows the simulation of 4 scenarios, which is a selection of a wide variety of phenomena than can occur during the operation of an engine.
An observation regarding the scenarios implemented in the simulation is the fixed intensity of the phenomena. For example, overpressure is simulated on three different levels by varying the multiplication factor applied to the pressure values, but it could be of any level above the design pressure values. The same statement is valid both for engine knock (as explained in the theory chapter) and for misfire, which can also be only a portion of the air-fuel mixture instead of a total misfire.
In this model, the generator is simplified using a shaft with inertia whose rotational speed is constrained on the free end by an external constraint. This approach is valid and very efficient but limits the range of applicability of the simulation only to cases where a big electricity grid is involved. In smaller environments, as for example the operation in the island mode (in which the electricity present in the grid comes only from the generator and is not influenced by other sources), the generator does not constrain the speed of the system and it is the control unit of the engine that is in charge of adjusting the engine load to maintain it constant.
Moving to physics related aspects, in this model the joint connections of main journals, connecting rod big-end and small-end bearings have been simplified. In the journals, the viscous speed-dependent friction coefficient due to the lubricant film is accounted for, and in the connecting rod bearings the links are simply frictionless perfect joints. This means that the model does not consider the vertical damping given by the thin oil film that is present in the bearings (Hother-Lushinson and Johnson 1963). In general, this approach is commonly accepted, but when the simulated phenomenon involves also high-frequency excitations, there is the risk that the said excitations get transmitted even if they could be heavily damped in reality. Figure 27 shows a comparison between the spectra of the flywheel speed (simulated) when the engine operates in normal conditions and with heavy knock occurring in cylinder number one. In case of knocking, it is possible to see (marked with a dashed circle) that the flywheel has excitations (of very small amplitude) in the same frequency range of the engine knock (visible in figure 1). It has not been verified if the same excitations are present also in the real engine, but most likely they are visible in the simulations only because the radial damping in the bearings is not included.

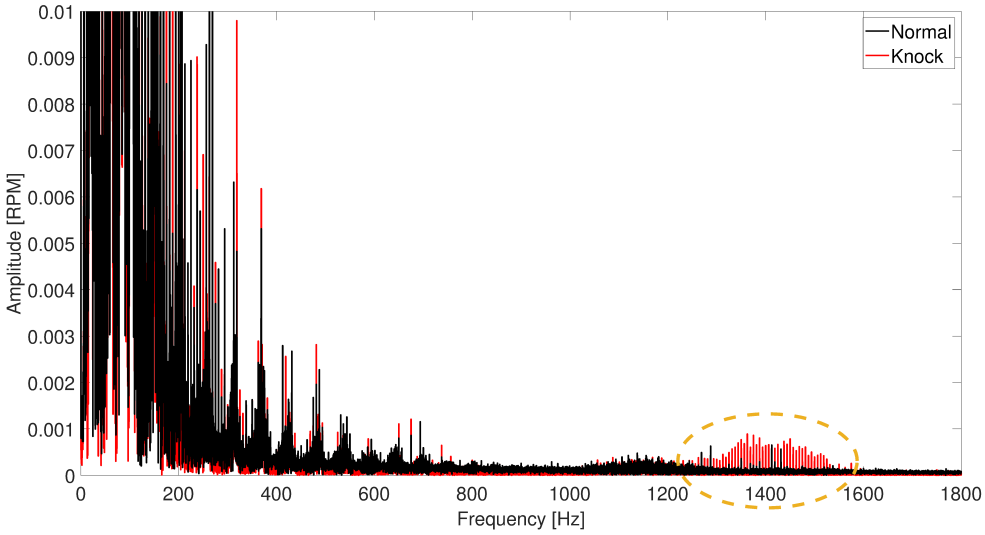

Figure 27: Effect of knock on the flywheel at high frequencies

3.6.2 Real engine

Bermeo setup The engine that is simulated is a test engine located in a test cell. This means that it operates in ideal conditions and all the parameters are monitored more accurately compared to what happens in reality. This guarantees consistency and reliability in the measurements but also creates an ideal situation that does not always reflect the real world. Of course, for the purpose of validating the model, it is beneficial to have stable conditions, but this also makes the simulation behave as an engine in ideal conditions.

Production engine setup Wärtsilä produces engines that are employed in different ways both in marine and industrial environments. The engine arrangement and setup are adjusted to match customer needs and the fuel type and quality. This, together with the small engine specific differences due to manufacturing tolerances, often creates unique situations in which each engine has a slightly different behavior. Clearly, this kind of simulation cannot account such aspects because the level of detail is not high enough. To do so, it would be necessary to use data measured directly from the field (at least to train the algorithms for the normal behavior) or to build more comprehensive and detailed simulations that are installation specific. Then, it would be also important to define all the anomalies that could occur in the real engine and train the algorithm to recognize them.

Measurement system A second aspect that would probably need to be implemented in the production engines is the measurement system. At the moment, the actual system described in paragraph 3.1.1 is capable of creating 1500 pulses/second at nominal speed (Storm et al. 2017), which according to the Nyquist theorem would allow to detect frequencies up to 750 Hz. In practice, for having an accurate

reading of the amplitudes of the constituent waves of a signal, the sampling frequency should be much higher than twice the maximum frequency of the study range. The maximum measurement error that can occur in sampling a signal is defined as

$$Max\ error\ [\%] = 100 \cdot [1 - \cos(\pi/N)] \tag{16}$$

and a representation of that equation is visible in figure 28. It is clear then, that

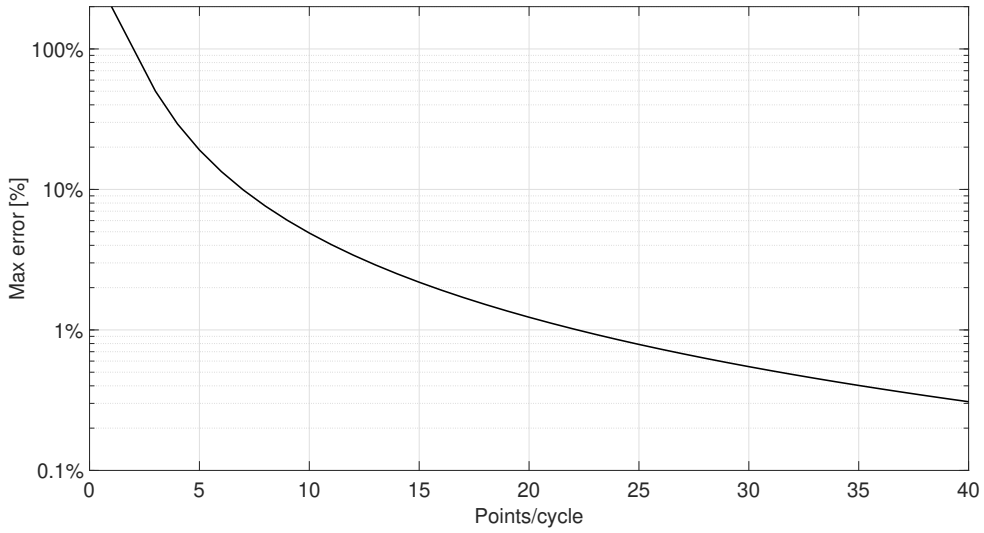

Figure 28: Maximum peak measurement error vs the number of sample points per period

with the actual measurement system, the constituent waves of a signal in the 0-200 Hz range are read with a maximum error of 8.6%. If we want to sample a signal in that range with no more than 1% error, we should have at least 4600 pulses/second, corresponding to 23 times the upper limit of the frequency range (Piersol and Paez 2010). It is therefore evident then that, even in case of transmission of the knock excitation, the actual measurement system is inadequate. At least 34500 pulses/second are needed to accurately detect the high-frequency oscillations.

3.6.3 Algorithms

Limitations The biggest limitation of these algorithms is that they require a big pool of data to perform a proper training. It is important to gather as many samples as possible to perform an accurate training as the more information they receive the more accurate the predictions become.
A second limitation that concerns ML is that the range of accuracy is limited to the one of the training range. For example, during this research, we feed to the algorithm data that are simulated at a constant speed and engine load. This makes the algorithm very accurate to evaluate spectra calculated in those conditions, but if we change the load, the algorithm ceases to be accurate as it does not know, in that

new situation, what is the expected behavior of the engine in the different scenarios. A third problem is that the algorithms that are the object of the study are not able to judge unknown scenarios. This underlines the importance of including as many anomalies and situations as possible in order to make sure that the algorithms do not find themselves incompetent of making an evaluation.
Most of kNN algorithms partly lose their effectiveness when the space becomes high-dimensional due to the curse of dimensionality. The concept of local neighbourhood loses its sense when there are hundreds of thousands of dimensions in the sample (in this case the number of peaks stored in each sample) due to the high noise that characterizes high-dimensional spaces. (Zhang 2013)
Artificial Neural Networks have been studied for several decades and with time the disadvantages of these algorithms have decreased. Nonetheless, in some cases approximations made with sigmoidal functions may converge slowly. (Svozil et al. 1997)

Capabilities The advantage of the kNN methods is that the algorithms do not rely on any assumed distribution of the data. This is really beneficial when the data is scattered and it is hard to define a hard boundary between the groups, as for example SVM algorithms try to do. (Zhang 2013)
For the multi-layer feed forward networks, the main advantages concern the learning phase, the non-linearity, the input-output mapping and the robustness. More in detail, the neural networks are able to adapt themselves without any action taken from the outside; the non-linearity of the neurons in practice allows the network to handle non-linear problems such as the differential equations of the torsional models. The ability to adapt the response of the network on the basis of the expected output represents a powerful feature of this family of algorithms which results in high robustness. As for the pattern recognition algorithms, also the ANNs increase their accuracy with increasing the amount of samples in the training phase. (Svozil et al. 1997)

3.7 Accuracy and possible sources of error

3.7.1 Simulation model

The torsional vibration model, for the abnormal cases, uses the same pressure curves in the faulty cylinder during the whole simulation time. This allows to create a set of samples for a given anomaly, occurring in a specific location, which are similar. However they still take into account the small variation generated by the remaining normal operating cylinders. In reality the intensity of the anomalies may varies between the cycles and this is not considered in the model. A solution could be to simulate the phenomena at different levels of intensity or to vary the intensity within the same simulation.
After the validation phase, the anomalies are reproduced by replacing the normal pressure traces with faulty ones with the assumption, generally correct, that the model will maintain its accuracy. However, the original system has a high complexity

and the simulation required some simplifications. This introduces the possibility that, depending on the cases, the model could generate an output which is not completely representative of the actual behavior. The only clarify this aspect is to validate the results with data recorded in the same conditions, but given the nature of the anomalies it might be difficult.

3.7.2 Data processing

The analysis of the results of the simulations is limited to the 0-200 Hz frequency range. The reason of this choice is dual: the most relevant peaks in the frequency analysis are located between those limits, the actual measurement system in use on the production engines is not effective at higher values. The machine learning algorithms compare the samples by corresponding peaks, which means that high and low peaks have the same influence. For this reason, also peaks at higher frequencies can be useful to perform the classification and including them in the samples would increase the amount of information available.

3.7.3 Algorithms

In data analysis, it is considered a good practice to scale the values of the samples between 0 and 1. Sometimes, this is also a requirement of the algorithm itself. In the case of a commercial tool, as the one adopted for this study, it is better to prepare the dataset before giving it as input, because there is less control on the way the algorithm handles the input data. In this case, however, the dataset has been tested with a scaling done both by column and globally, showing no variation in the results. The Classification Learner offers the possibility to choose how to handle the validation phase to prevent the overfitting of the model. The options are a cross-validation that divides the dataset into folds and evaluate them separately, a holdout validation which uses part of the samples for a single validation, and no validation. For this study a cross-validation with 5 folds is chosen. The method is recommended for small datasets because it performs several trainings using the out-of-fold samples and keeps the in-fold data to asses the performance. A higher amount of folds reduces the likelihood of an overfit but decreases the amount of samples available to assess the accuracy, whereas a smaller amount of folds guarantees a more reliable evaluation because the validation sets contain more elements.
The Neural Net Pattern Recognition tool allows to decide the amount of hidden neurons involved and how the dataset is divided between training, validation and testing groups. The default values are 10 neurons , 70% for the training and 30% between the remaining two groups. They represent a standard starting point, but these parameters affect the performance of the network and require additional testing to evaluate the potential increase in accuracy or error.

4 Results

4.1 Post-processed simulation results

The simulation results consist of values of the crankshaft speed measured at the flywheel and stored in a time domain.
Data in a time domain in certain cases is easy to interpret, especially when it is possible to compare it with a reference value recorded in normal conditions. In

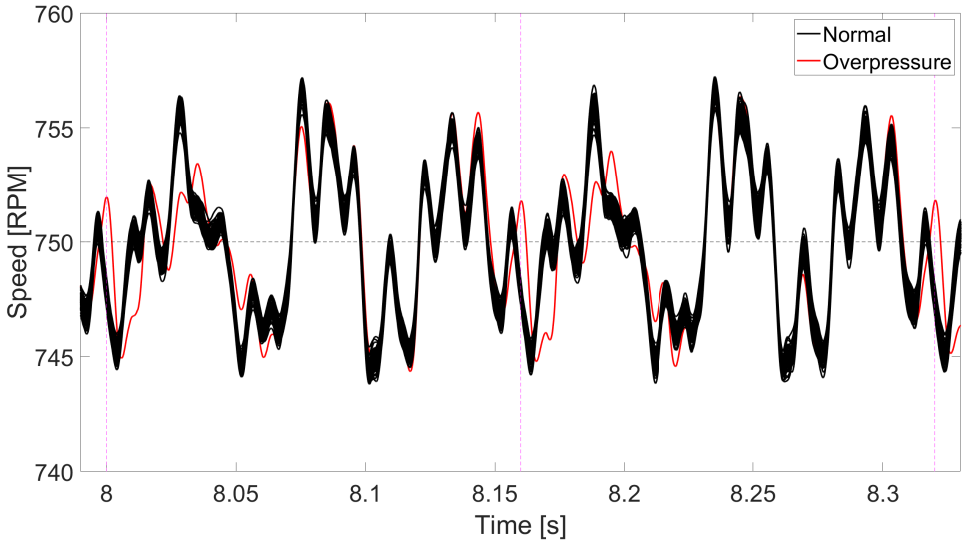

Figure 29: Comparison of the flywheel speed over two engine cycles between normal situations and overpressure in cylinder 1

figure 29 there is a plot of all the values of speed for the normal cases together with one case where overpressure in cylinder 1 is simulated. It is clear how, in normal conditions, the speed has a very consistent behavior through the whole domain. The overpressure, in this case, is easily detectable, and it is also possible to identify which is the faulty cylinder: around the TDC of the overpressuring cylinder, around $t = 8\ s$ and $t = 8.16\ s$, first the crankshaft slows down compared to a normally behaving engine, then the speed increases in value and the gap goes to zero in approximately 270 crank angles (corresponding to 0.06 seconds). If we look at figure 30, we have the same plot as before but this time with heavy knock instead of overpressure. It is possible to see a similar behavior as before, but it is mainly due to the slight overpressure that is generally associated with the heavy knock. This suggest that, at glance, it is possible to detect that we have an abnormal situation but also that it would be easy to misinterpret the two phenomena.

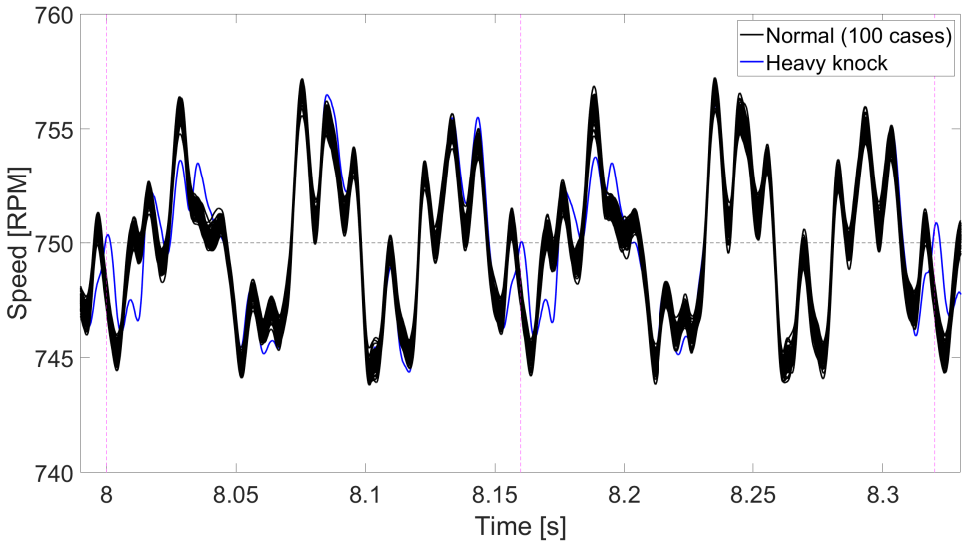

Figure 30: Comparison of the flywheel speed over two engine cycles between normal situations and heavy knock in cylinder 1

Figure 31 shows the speed of all the normal and overpressure cases simulated. They are 100 and 60, respectively. Figure 32 shows the same for the normal and knocking cases. It stands out from those pictures that an analysis in the time domain can be quite complicated, especially if the phenomena generate similar behaviors as shown in figures 29 and 30.
The processing of the results of the torsional vibration simulations aims to convert the speed signal from a time domain to a frequency domain and to clear the spectra

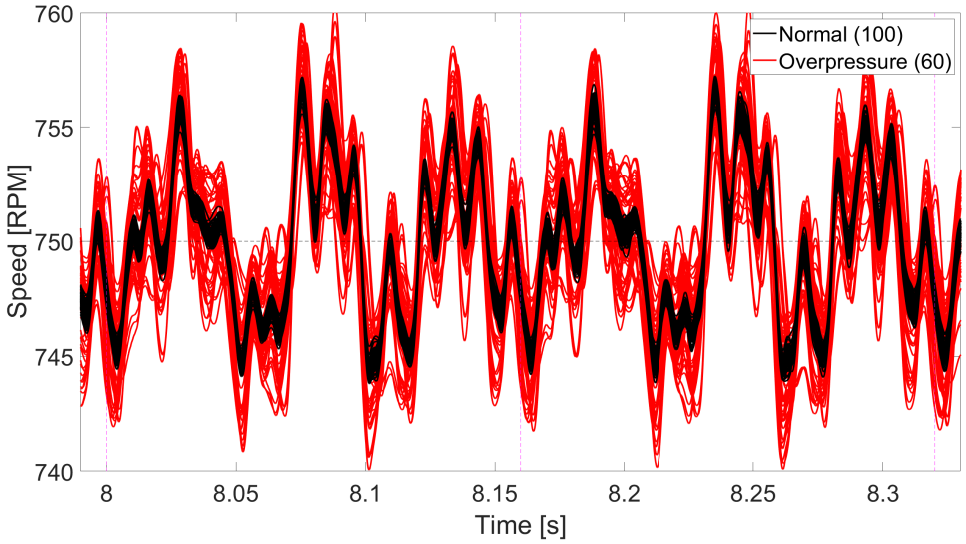

Figure 31: Comparison of the flywheel speed over two engine cycles between normal and overpressure situations

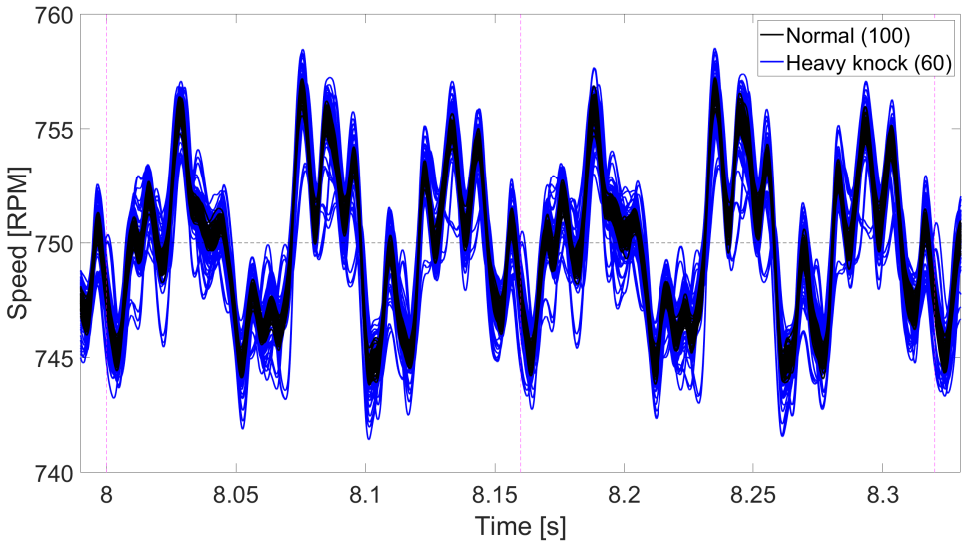

Figure 32: Comparison of the flywheel speed over two engine cycles between normal and heavy knock situations

of the 7000 subsets from the irrelevant values. The objective of this is to obtain a dataset which is lean and contains only relevant values. A visual clarification of the obtained dataset is visible in figure 33 and larger in figure 34. The figures show a selection of 25 cases per scenario. Observing the images, it is clear how the behavior in normal condition is consistent among all the subsets and how each scenario has a behavior that differs from the others although some scatter is present. This scatter, in practice, makes it almost impossible to detect, just by observing the peaks by eye, to which scenario a random spectrum belongs to.

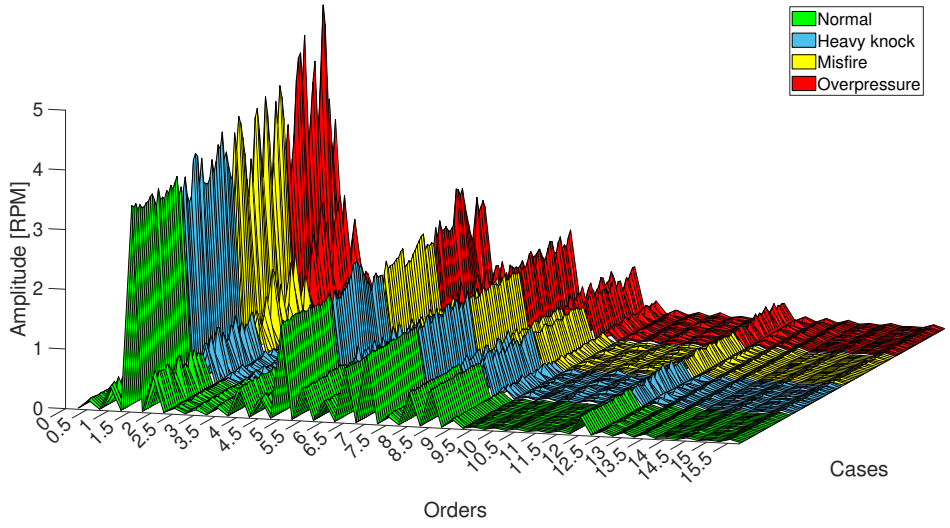

Figure 33: Post-processed spectra of a selection 100 different cases (25 per scenario)

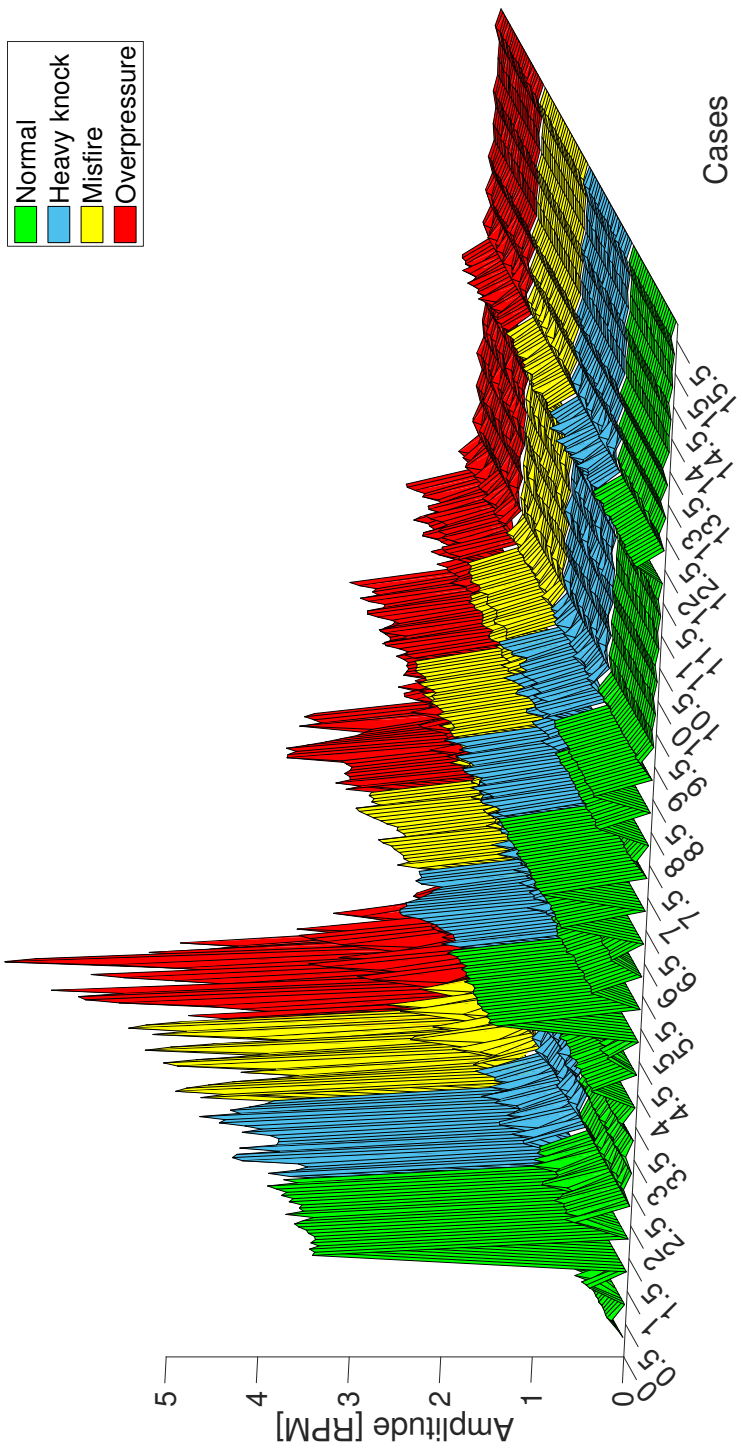

Figure 34: Post-processed spectra of a selection 100 different cases (25 per scenario)

4.2 Pattern recognition algorithms

As explained in the previous chapters, the Classification Learner application of MATLAB allows to evaluate the accuracy of several pattern recognition algorithms and then to generate a code to be used with new samples. All the tested algorithms

Table 2: Performance of the tested pattern recognition algorithms

n	Algorithm	Accuracy
1	Tree (max 100 splits)	89.8%
2	Tree (max 20 splits)	80.5%
3	Tree (max 4 splits)	69.7%
4	Linear discriminant	82.2%
5	Quadratic discriminant	95.5%
6	SVM (linear)	91.3%
7	SVM (quadratic)	96.8%
8	SVM (cubic)	96.2%
9	SVM (Gaussian fine)	85.7%
10	SVM (Gaussian medium)	96.7%
11	SVM (Gaussian coarse)	92.5%
12	kNN (100 neighbors)	95.1%
13	kNN (10 neighbors)	95.1%
14	kNN (1 neighbor)	68.3%
15	kNN (10 neighbors and cosine distance metric)	95.2%
16	kNN (10 neighbors and cubic distance metric)	94.3%
17	kNN (10 neighbors and weighted distance metric)	95.8%
18	Ensemble (Boosted Trees)	88.0%
19	Ensemble (Bagged Trees)	96.2%
20	Ensemble (Subspace Discriminant)	78.5%
21	**Ensemble (Subspace kNN)**	**97.0%**
22	Ensemble (RUSBoosted Trees)	82.2%

with their configurations and the calculated accuracies are listed in table 2. A brief description of the algorithms:

- The Tree algorithm (1 to 3) performs binary decisions on the basis of weights tuned during the training phase.
- Discriminant analysis (4 and 5) assumes that different classes generate data on the basis of different Gaussian distributions.
- SVM (6 to 11), described in paragraph 2.4.1 of the theory chapter, is tested with different methods to evaluate the Gram matrix, which is used to transform non-binary problems into binary situations, so that the algorithm can correctly define the boundary hyperplanes.

- kNN (12 to 17), is tested with different quantities of nearest neighbors and different ways to calculate the distance metrics. The cosine distance metrics, for example, is defined as:

$$d_{st} = \left(1 - \frac{x_s y_t'}{\sqrt{(x_s x_s')\,(y_t y_t')}}\right) \tag{17}$$

  where x_s and y_t are the vectors containing the coordinates of the samples in the domain.

- Ensemble methods (18 to 22) combine multiple classifiers of the same family to improve the accuracy of the prediction (Gul et al. 2016).

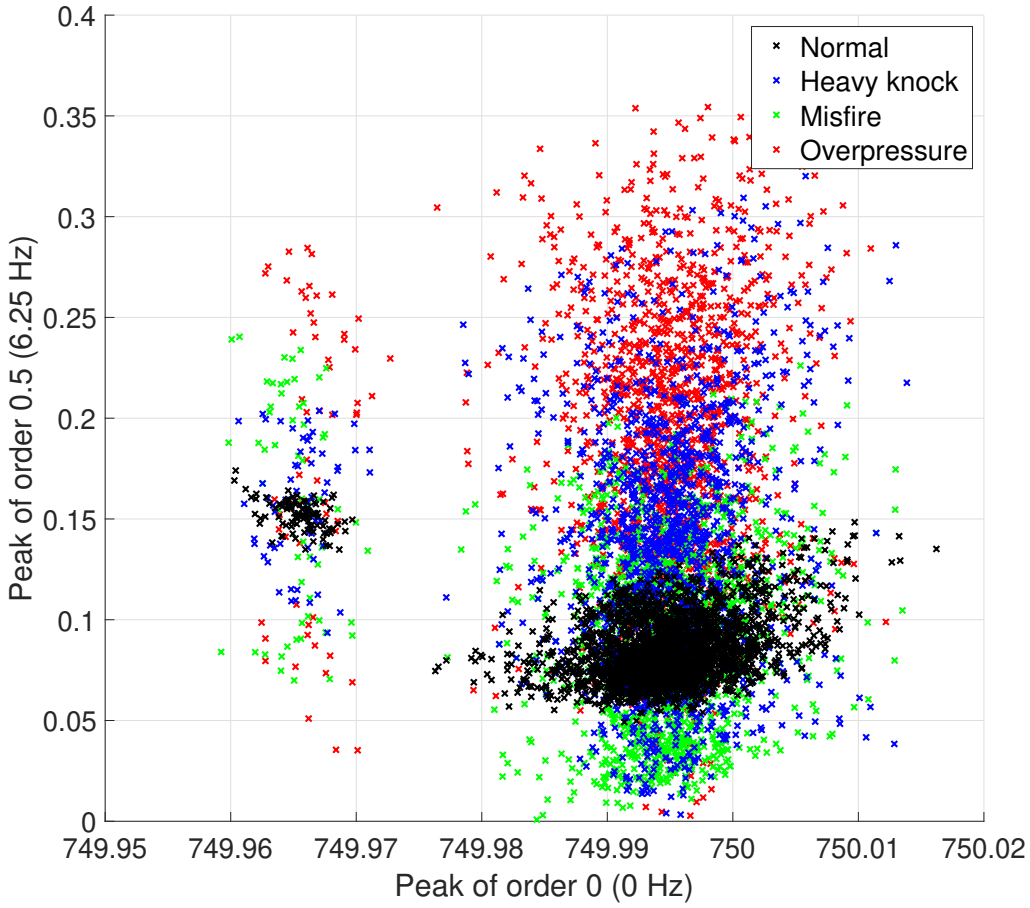

Figure 35: Scatter plot of 2 dimensions (speed and order 0.5)

Figure 35 shows a representation of two dimensions of the dataset, in this case the average speed (on the x-axis) and the firing order (y-axis). As for the 3D representation of the spectra, also here the normal samples (black marks) tend

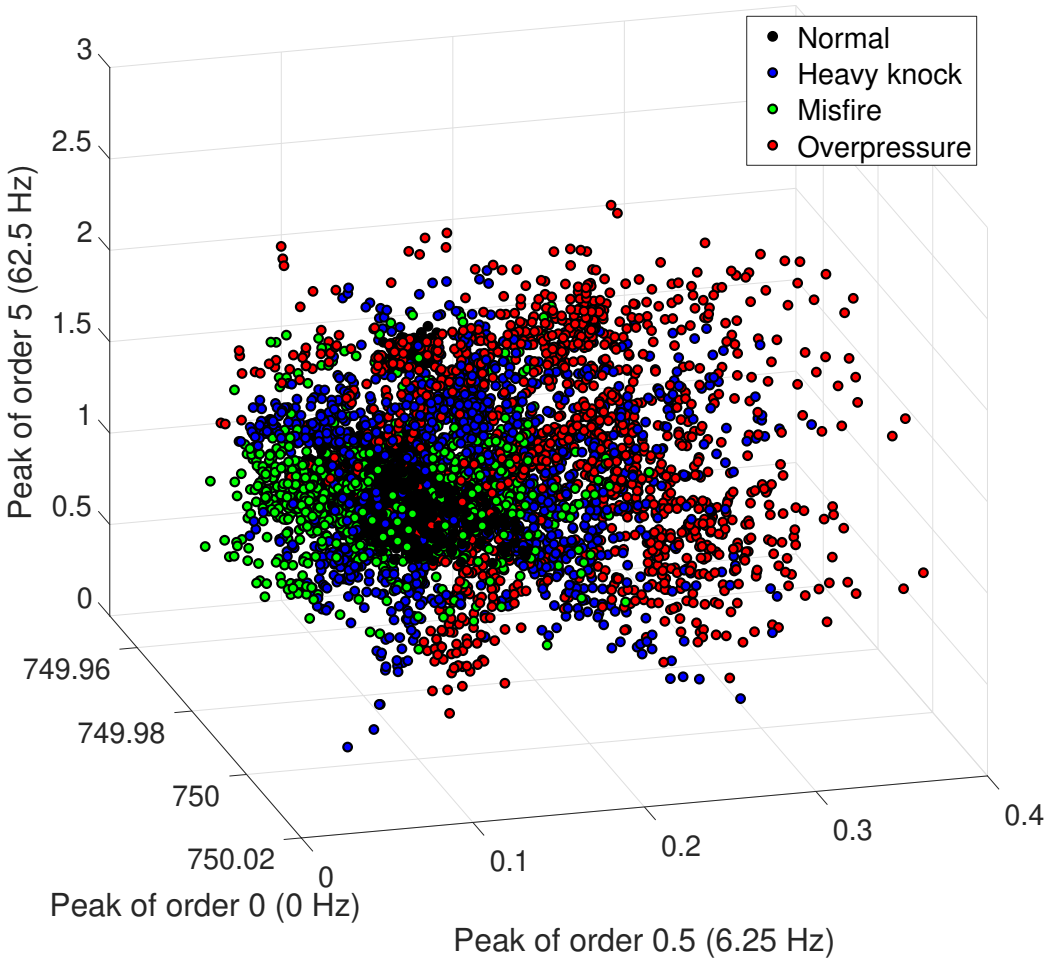

Figure 36: Scatter plot of 3 dimensions (speed, order 0.5, order 5)

to group in a certain area. The misfires and the knockings find their locations respectively below and above the normal cases due to the lack and excess of power in the combustion cycle. Overpressure cases are more sprinkled in the speed dimension due to to the higher amount of energy required for the compression phase that slows down the crankshaft and the consequent spring effect that accelerates it. Nevertheless, the combustion order shows higher values compared to the previous scenarios.
In figure 36, a similar representation is visible but with a third dimension included. There, it starts to be more complicated to define the boundaries between the scenarios. Nonetheless, the normal cases still find themselves in a quite delimited subspace of the domain.
In figure 37, there is a visual representation of the performance of the algorithm 21 "Ensemble (Subspace kNN)" which is the one that has the highest accuracy among all the tested algorithms. The figure shows a double-entry table with the true class

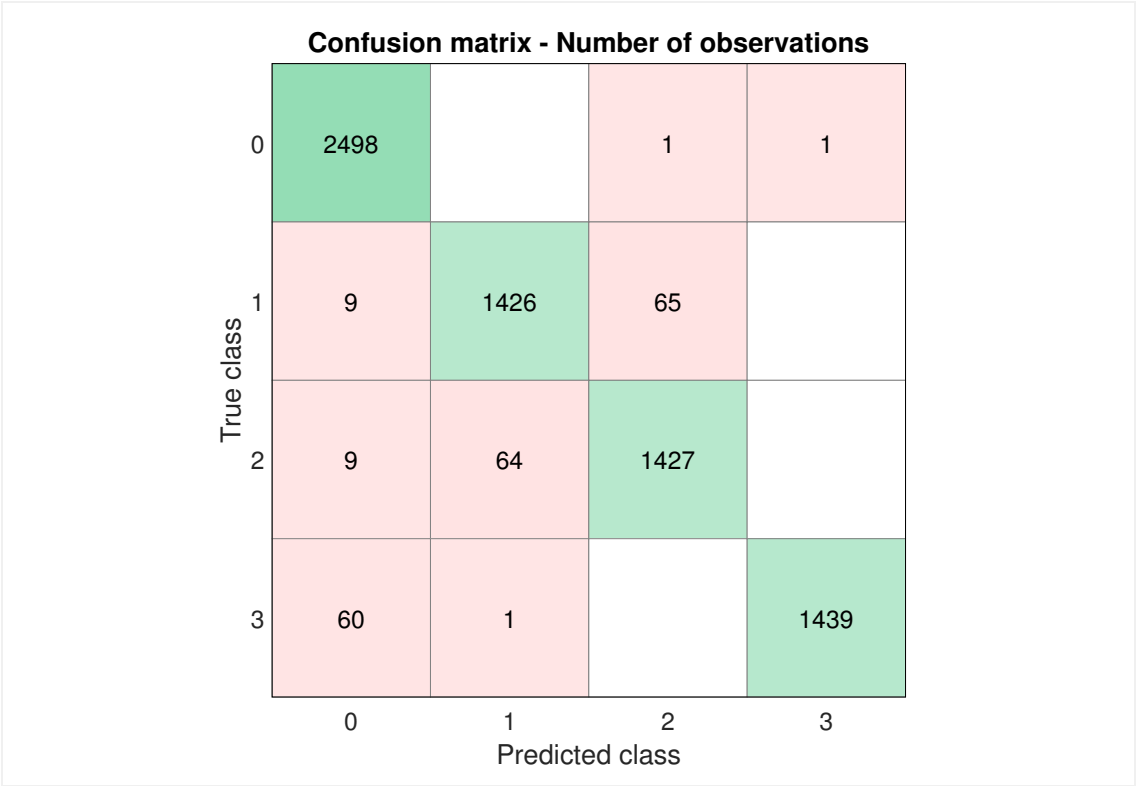

Figure 37: Confusion matrix summarizing the output of the Classification Learner App

of each analyzed sample and the class that the algorithm predicted. The scenarios are number-coded as

- 0 for normal behavior,
- 1 for heavy knock,
- 2 for misfire,
- 3 for overpressure.

The same results are also visible in the form of true and false positive observations rates in figure 38. There, the values are in percentage and thus the accuracy of the algorithm stands out more clearly. (*Statistics and Machine Learning Toolbox™ User's Guide* 2018)

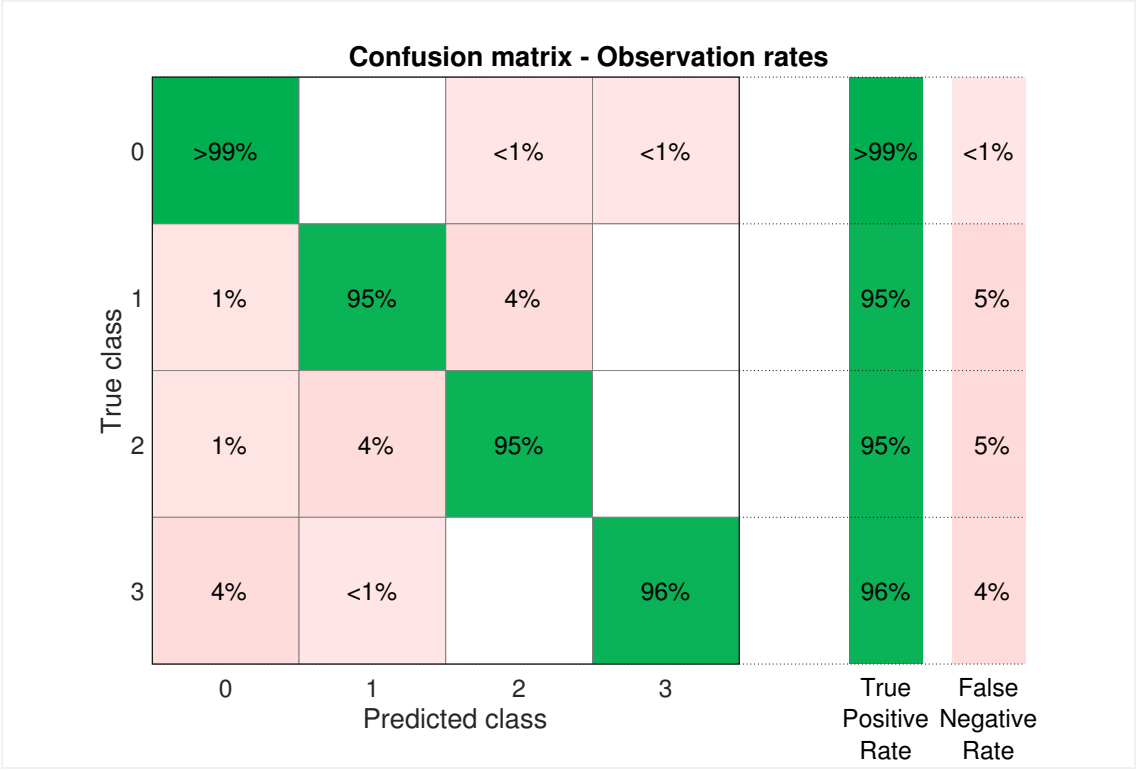

Figure 38: Confusion matrix summarizing the output of the Classification Learner App

4.3 Artificial neural network

The Neural Net Pattern Recognition application allows to train a 2-layer neural network, test its accuracy and deploy it for the use with new samples.
Table 3, gives a summary of the main figures related to the process. The three

Table 3: Properties of the simulation output

Group	Samples	Cross-Entropy error	Fraction of samples misclassified [%]
Training	4900	1.985	4.367
Validation	1050	5.675	6.286
Testing	1050	5.604	6.190

groups training, validation and testing are evaluated for the cross entropy and for the fraction of samples misclassified independently. The most relevant group, though, is the testing as it represents an assessment of the trained network and should reflect the performance of the network when tested with new samples. In figure 39, there

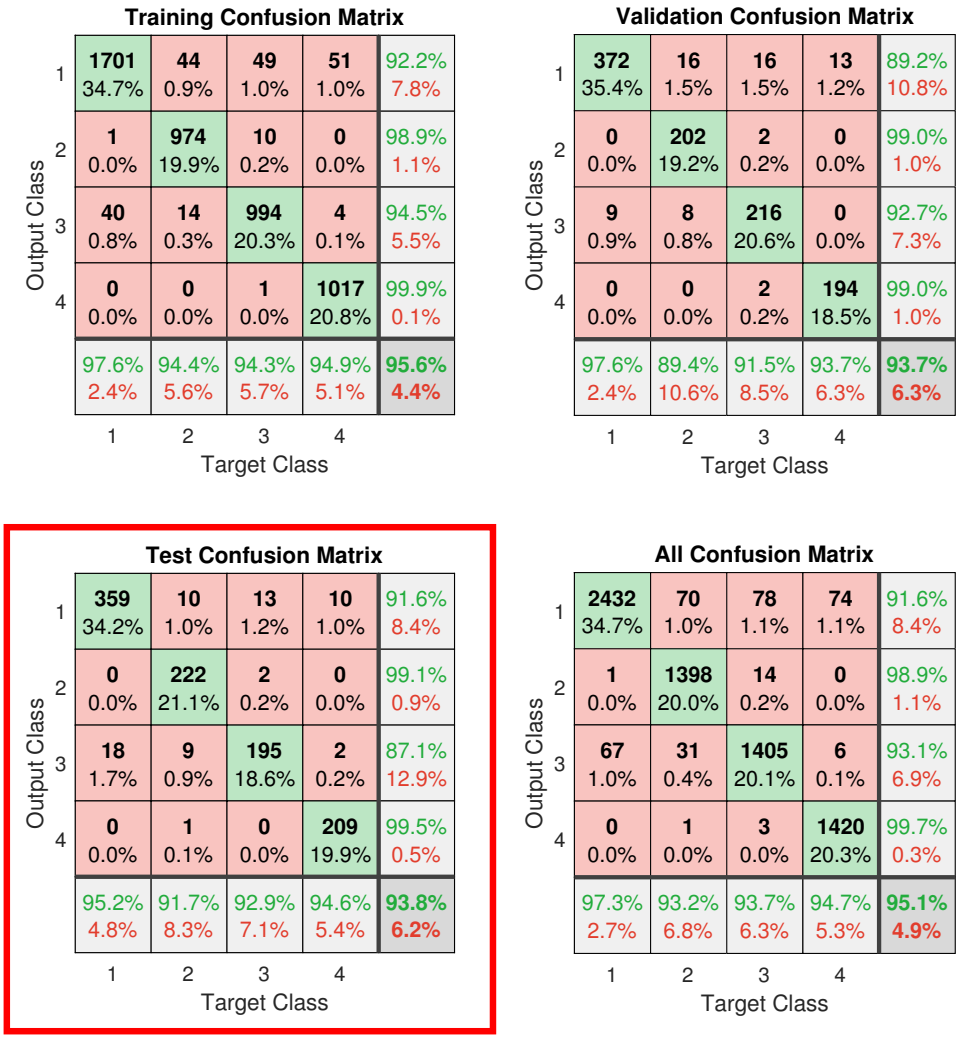

Figure 39: Confusion matrices summarizing the output of the Neural Net Toolbox

are the confusion matrices of the three groups and a fourth with the sum of the previous ones. The number-coding of the scenarios, called classes in the machine learning terminology, differs slightly from the previous:

- 1 for normal behavior,
- 2 for heavy knock,
- 3 for misfire,
- 4 for overpressure.

When the confusion matrices are read by line, the last columns show the rates of true positive and false negative classifications. Conversely, if read by columns, the

last lines show the positive predictive values and the false discovery rates.
Figure 40 shows the Receiver Operating Characteristic curves (ROC) for the groups in which the dataset is divided and the overall ROC curve. The ROC curve is a

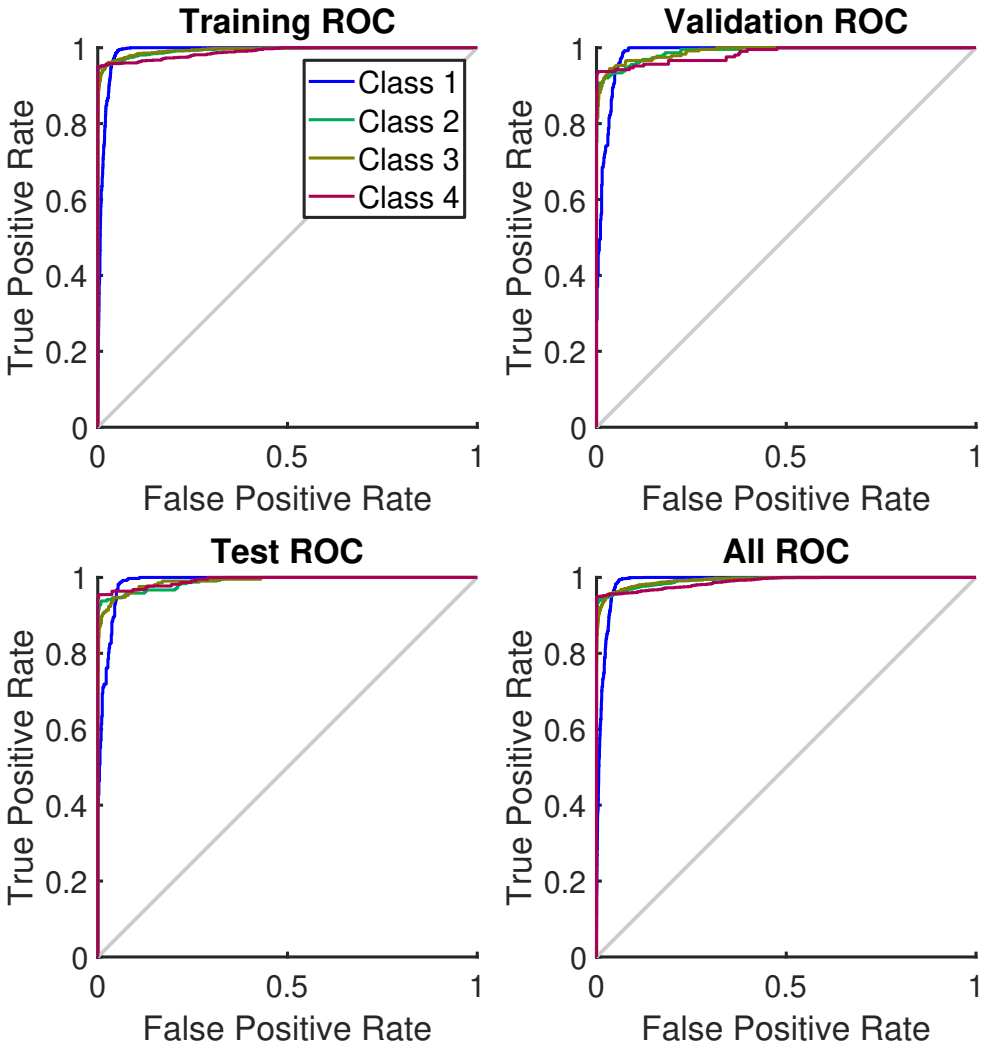

Figure 40: Receiver Operating Characteristic of the neural network training

diagnostic tool used to asses the performance of the classifier that compares the false positive rate (FPR) with the true positive rate (TPR) (Beale et al. 2018a). The 45^o lines represent a threshold and the curves should stay above them. In this case, all the classifiers are behaving in a healthy way.

5 Discussion

The aim of this work is to understand whether it is possible to adopt novel methods based on machine learning and data mining to detect unwanted behaviors in the operation of a medium speed engine.
This project originates from the will to increase the reliability of the safety systems embedded in Wärtsilä engines and from the customers' request to comply with the strict SIL-2 risk-reduction level defined in the IEC 61508 international standard on functional safety.

Torsional vibration simulation The torsional vibration model adopted is very versatile, and can be used to generate unlimited outputs for virtually any kind of scenario. When the solver runs on a modern multi-core cluster, a simulation of 60 engine cycles takes approximately 3.5 hours, which is a huge improvement compared to the time and money that creating and measuring the same anomaly on a real test engine would require.
The model reproduced the behavior of a real engine accurately, without the need to include 3-D FEM simulations, which would increase the complexity significantly. However, the validation phase highlighted the sensitivity of the model to the values of the coupling parameters. This indicates that the results should be interpreted with caution and validated with field data whenever possible.

The three anomalies studied are described in the theory chapter. Their respective inputs to the model, in the form of pressure traces, are visible in figures 1, 4 and 5. It is worth noting that those three curves are very simple to identify when compared to a reference curve. However, when they become part of a complex system like a V20 engine, their detection becomes much more difficult, and calls for data analysis tools.

An analysis in the time domain clearly highlights abnormal situations. When the engine operates in normal conditions, its behavior is very consistent, as can be seen in the torsional vibration analysis results in figures 31 and 32. Therefore, it is easy to spot trends that lie out of the normal operation range. However, only a binary detection between normal or abnormal behavior is possible. In order to gather more information on the anomaly, algorithms that can perform classifications are needed.

Patter Recognition Algorithms The pattern recognition algorithms have different levels of accuracy, depending on which algorithm is chosen but also on different setups for the same algorithm (table 2).
The highest accuracy, 97%, is achieved with an Ensemble based on the k-nearest neighbors method. The confusion matrix in figure 38 shows that the algorithm can be very reliable in detecting normal situations, with almost no false negatives. This aspect is very important for safety-related applications, as it would avoid unnecessary interruptions to the operation of the system. When the anomalies are concerned,

4% of the cases of overpressure and 5% of the cases of both misfire and heavy knock are misclassified. Interestingly, the misclassifications happen mainly between heavy knock and misfire classes, and do not involve normal behavior or overpressure classes, which means that the algorithm is still detecting an anomaly, albeit not of the right type. Considering that the algorithm is trained on a fairly limited amount of samples (7000 in total) and the settings are not fine-tuned, its overall performance is very promising and can be further improved.
The SVM and kNN algorithms also provide interesting results, with accuracies ranging between 85 and 97%. The only kNN that yielded a lower accuracy was limited by having only 1 neighbor available for the classification; simply increasing the number of neighbors to 10 was enough to boost the accuracy to the top part of the chart. No further improvement was achieved increasing the number of neighbors above 10.
The discriminant algorithm, which is considered quick-to-train, has a good accuracy only when it involves a polynomial equation, as a linear approach is not able to capture the complexity of this specific problem.
At the bottom of the chart, with the lowest performance, there is the Tree algorithm. When the number of splits is equivalent to the number of classes, the accuracy reaches only 69.7%. Increasing the number of splits brings the accuracy closer to 90%. The results may improve with a higher number of splits, but the accuracy gap with the other algorithms at this stage suggests that they perform better in any case.

Artificial Neural Network The application used to assess the effectiveness of the 2-layer feed forward artificial neural network provides confusion matrices for all the phases of the learning process (figure 39).
When the focus is on predictions, the most interesting is the confusion matrix of the test group of samples. It represents a post-training evaluation of the ability of the network to classify future cases. The overall accuracy of the neural network, 93.8%, is slightly lower than that of the Ensemble kNN algorithm, even though in some learning sessions an accuracy of 95.9% was achieved. The accuracy changes because the weights are assigned to the neurons at the beginning of the training phases randomly, and the division of the samples into the three groups (training, validation, test) is also random. As a consequence, the iterative learning process develops towards different weight values for the neurons. At this stage it is impossible to obtain an unchanging level of accuracy from run to run.
However, in all the trainings of the network, the test samples of heavy knock, misfire and overpressure were correctly identified but the classification of normal samples was occasionally inaccurate. This suggests that the neural network is more precise than the pattern recognition in distinguishing the anomalies rather than the normal cases.

In conclusion, both the pattern recognition algorithms and the neural network show promising levels of accuracy. At this stage, the best-performing pattern recognition algorithm is more reliable in the distinction between normal and abnormal cases. This is in agreement with the known tendency of this kind of algorithms of effectively solving binary problems, while encountering more difficulties in multi-class

situations where the boundaries between classes are not clearly defined. On the other hand, the neural network is more precise in classifying the anomalies, exactly where the pattern recognition algorithm has limitations.

Future improvements

This research work lays the foundation for a novel approach that promises interesting results. As all the things which are at the beginning, several steps are needed at this point. Here is a list of the main actions that are required to further develop this method:

Increase the complexity and the diversity of the dataset. The dataset used for this initial assessment is limited to a specific engine speed (750 RPM) and load (~96%). This makes the actual trained algorithms and network unable to evaluate other situations:

- Multiple anomalies. Knock, misfire and overpressure are three common anomalies that occur in combustion engines, but there are several others that can occur and that should be included in the training. A suggestion would be to list the anomalies that are most likely to take place and include them in the training of the algorithms.
- More variety in the simulated scenarios. All the anomalies can occur with different intensities and thus they affect more or less heavily the behavior of the engine. Including different levels of magnitude would increase the accuracy of the algorithms, especially in the areas of the domain close to the boundary regions where the algorithms have the hardest time in deciding to which class the samples belong to.
- More scenarios. The engine can operate at different load levels and speeds depending on the application. It is fundamental that the training includes samples obtained in the same load and engine speed in which the engine will operate, otherwise the algorithms will not be able to assess correctly. To include more operative conditions, a possibility would be to include in the training samples that come both from the simulations and from data acquired from existing installations.

Sensitivity analysis of the variables affecting the algorithms performances. The machine learning algorithms have several parameters and setups that can affect their performance. An investigation of the effect of those elements is required to improve their performance.

Combination of the pattern recognition algorithm and the neural network. The tested PRA and NN are effective on different aspects of the classification. The two approaches could then be combined together. The pattern recognition algorithm could be trained only to identify the abnormal situations, leaving to the neural

network the task of labelling the anomalies. In this way, the strong points of both approaches are exploited and the overall accuracy would increase.

Validation of the anomalies with field data. The GT simulation shows a high level of accuracy but also the need of a fine tuning to be accurate. A validation with real data acquired from the field would increase the reliability of the approach. Figure 41 illustrates how the data from an engine can be used to validate the torsional vibration model and to increase the dataset of samples.
Collecting data from different existing systems could be an effective way to raise the level of accuracy and to account for the small variations that are installation-dependent.

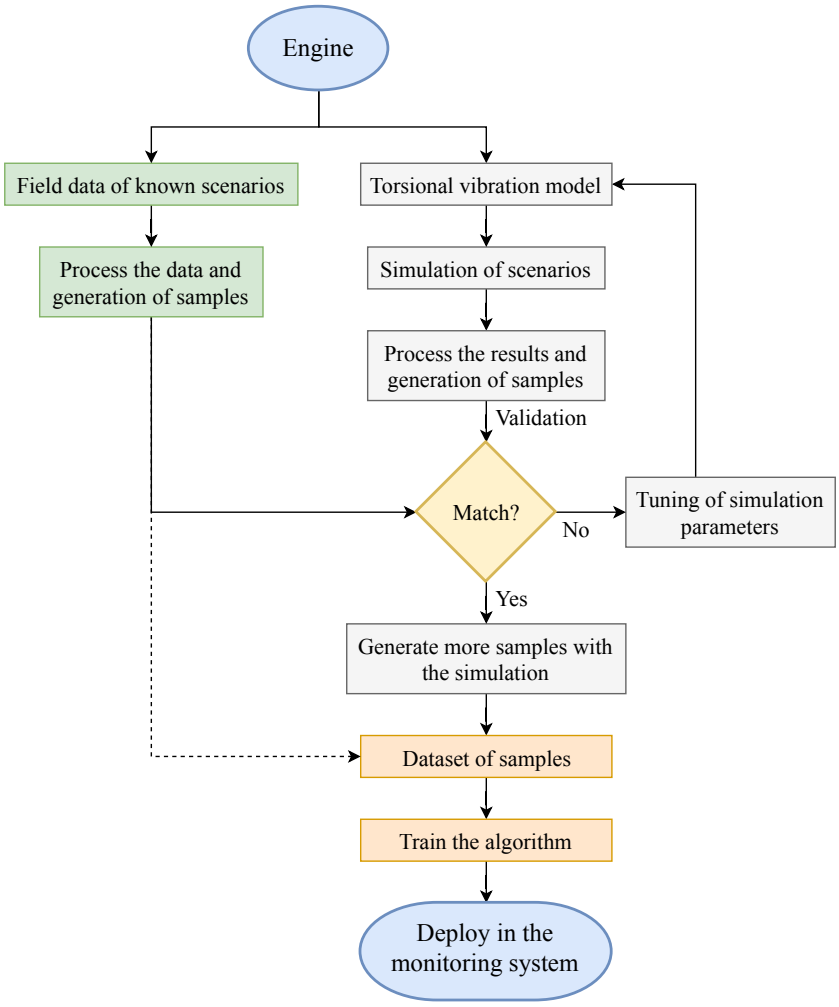

Figure 41: Flow chart illustrating the process of validation with field data and the creation of the dataset of samples

6 Conclusions

This research project comes from the need of more reliable anomaly detection systems for the engines produced by Wärtsilä. The aim was to understand if there is a possibility to reduce the amount of sensors involved in the detection loops with the goal of decreasing the possibilities of a failure and therefore comply with the SIL-2 requirements of IEC 61508.
For this reason, a suitable location for a sensing system was identified and the following research question was placed:

Is it possible to detect ab-normal situations only by analyzing the speed of the crankshaft?

The starting point to answer that question is building a torsional vibration model of a real engine and validating it both with data acquired from the field and by assessing its behavior when the input changes. The model is then used to generate flywheel speed records for 4 different scenarios: normal conditions, heavy knocking, misfiring and overpressure.
By studying the speed records, it is possible to see a strong affinity between the normal samples, and this makes possible, to some extent, the detection of an abnormal situation over a normal operation. Since this is not enough to also classify the cases, there is a necessity of a more advanced approach based on machine learning algorithms.
The focus in this phase is on two families of algorithms: pattern recognition algorithms and neural networks.
For the first family, the focal point was mainly on the Ensemble kNN, an algorithm that combines several classification models based on the k-nearest neighbors method to increase the predictive performance. The outcome was a high level of accuracy in the classification (97%) and a particularly low rate of false negatives (<1%).
The neural network chosen was a simple 2-layer feed-forward network, one of the first concepts of artificial networks that has been deeply studied and improved over several decades. Without performing a fine tuning of the network, it was already possible to achieve a remarkable level of accuracy (95.9% in the best cases), suggesting that there is room for improvement. The network turned out to be more capable than the PRA to discern the different anomalies but a bit less performant in the distinction between them and the normal cases.
Overall, both the approaches show already at this stage interesting classification capabilities and room for further refinement, for example, by analyzing a wider range in the spectrum or including measurements from additional sensors.

To answer the research question, the conclusion of this study is that, at a glance, monitoring the crankshaft speed seems a promising approach to detect misbehavior in the engine.
Simulation results are preliminary assessed from an acknowledged point of view, showing beforehand identifiable peculiar trends. In parallel with the understanding

of the phenomena, data-driven approaches, like the spectrum peaks isolation, are adopted to improve the quality of the anomalies detection. Eventually, the tested machine learning algorithms show that it is possible to recognize different scenarios after a training with a reasonable number of samples.

References

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, *46*(3).

Beale, M. H., Hagan, M. T., & Demuth, H. B. (2018a). *Neural network toolbox^{TM} reference*. The MathWorks, Inc.

Beale, M. H., Hagan, M. T., & Demuth, H. B. (2018b). *Neural network toolbox^{TM} user's guide*. The MathWorks, Inc.

Bishop, C. (2006). *Pattern recognition and machine learning*. Springer-Verlag New York.

Cavina, N., Corti, E., Minelli, G., & Serra, G. (2002). Misfire detection based on engine speed time-frequency analysis. *SAE Technical Paper 2002-01-0480*. doi:10.4271/2002-01-0480

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*. doi:10.1145/1541880.1541882

Gassner, G., Cenini, P., Ulstein, K. O., & Contessi, C. (2016). World's first ethane-powered marine vessels. *Wärtsilä Technical Journal*, *2*, 38–43.

Gul, A., Perperoglou, A., Khan, Z., Mahmoud, O., Miftahuddin, M., Adler, W., & Lausen, B. (2016). Ensemble of a subset of knn classifiers. *Advances in Data Analysis and Classification*. doi:10.1007/s11634-015-0227-5

Hairer, E., & Wanner, G. (1996). *Solving ordinary differential equations ii*. doi:10.1007/978-3-642-05221-7

He, X., Qi, Y., Wang, Z., Wang, J., Shuai, S., & Tao, L. (2015). Visualization of the mode shapes of pressure oscillation in a cylindrical cavity. *Combustion Science and Technology*, *187*(10), 1610–1619. doi:10.1080/00102202.2015.1038385

Hertz, H. (1881). Über die berührung fester elastischer körper. *Journal für die reine und angewandte Mathematik*, *92*, 156–171.

Heywood, J. (1988). *Internal combustion engine fundamentals*. McGraw-Hil.

Hother-Lushinson, S., & Johnson, S. (1963). Damping properties of thin oil films subjected to high-frequency alternating loads. *Journal of Mechanical Engineering Science*, *5*(2), 175–181.

Hsu, C.-W., & Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on neural networks*, *13*(2).

Ince, D. (Ed.). (1992). *Mechanical intelligence, volume 1 (collected works of a.m. turing)*.

Jun, H., & Moraga, C. (1995). The influence of the sigmoid function parameters on the speed of backpropagation learning. In J. Mira & F. Sandoval (Eds.), *From natural to artificial neural computation* (pp. 195–201). Springer Berlin Heidelberg.

Mohiuddin, A., Rahman, A., & Fikri, A. (2008). Design and development of valve train system for gasoline engine using gt-suite software. *International Journal of Mechanical and Materials Engineering*, *1*, 80–89.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of machine learning*. MIT Press.

Nyce, D. S. (2004). *Linear position sensors: Theory and application*. Wiley-Interscience.

Piersol, A. G., & Paez, T. L. (2010). *Harris' shock and vibration handbook* (6th ed.). McGraw-Hill.

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classfi ers* (pp. 61–74). MIT Press.

Sen, A. K., Litak, G., Yao, B.-F., & Li, G.-X. (2010). Analysis of pressure fluctuations in a natural gas engine under lean burn conditions. *Applied Thermal Engineering*, *30*, 776–779. doi:10.1016/j.applthermaleng.2009.11.002

El-Shahat, A. (2018). *Advanced applications for artificial neural networks*. doi:10.5772/intechopen.68505

Statistics and machine learning toolbox^{TM} user's guide. (2018). The MathWorks, Inc.

Storm, X., Salminen, H. J., Virrankoski, R., Niemi, S., & Hyvonen, J. (2017). Analysis of cylinder pressure measurement accuracy for internal combustion engine control. *SAE Technical Paper 2017-01-1067*. doi:10.4271/2017-01-1067

Sundsten, M. (2013). *A control arrangement for an inlet valve in a piston engine*. WO 2011/004059 (13.01.2011 Gazette 2011/02).

Svozil, D., Kvasnička, V., & Pospíchal, J. (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems*, *39*, 43–62.

van Basshuysen, R., & Schäfer, F. (2007). *Modern engine technology from a to z*. SAE International.

Walcher, H. (1994). *Position sensing: Angle and distance measurement for engineers - 2 rev.ed*. Butterworth-Heinemann Ltd.

Wang, Z., Liu, H., & D. Reitz, R. (2017). Knocking combustion in spark-ignition engines. *Progress in Energy and Combustion Science*, *61*, 78–112. doi:10.1016/j.pecs.2017.03.004

Wang, Z., Liu, H., Song, T., Qi, Y., He, X., Shuai, S., & Wang, J. (2014). Relationship between super-knock and preignition. *International Journal of Engine Research*. doi:10.1177/1468087414530388

Yu, L., Wang, S., & Lai, K. K. (2007). *Foreign-exchange-rate forecasting with artificial neural networks*. Springer.

Zbigniew, S. (2014). Intake valve and combustion chamber deposits formation – the engine and fuel related factors that impacts their growth. *Nafta-Gaz*, (4).

Zhang, J. (2013). Advancements of outlier detection: A survey. *ICST Transactions on Scalable Information Systems*, *13*, 1–26.

Appendix

The MATLAB code of the script used to process the results of the simulations. The outputs of the script are the matrices needed by the Classification Learner and the Neural Net Pattern Recognition apps.

```
%% 1. Load simulation results
% These workspaces contain the output of the torsional vibration
% simulations.
% The speed of the flywheel is stored in the variable
% X_FWspeed, where X changes according with the scenario: N for
% normal, O for overpressure, M for misfire, K for heavy knocking.
% Every variable has 160000 rows and as many columns as the number of
% cases simulated. The sampling rate is 20000 Hz thus each column
% contains 8 seconds of simulation data.

load('Simulation_data_TIMEsteps/NormalCases.mat')
load('Simulation_data_TIMEsteps/OverpressureCases.mat')
load('Simulation_data_TIMEsteps/KnockCases.mat')
load('Simulation_data_TIMEsteps/MisfireCases.mat')

%% 2. Generation of subsets of 2 cycles, FFT analysis and data arrange
% Compute spectra of the first 2 cycles of every case and store it by
% column. Then continue with the next 2 and so on until all the 50
% cycles are processed.

% There are 100 normal cases so the final matrix N_spec will be of
% 2500 columns.

N_spec=nan(3201,100*25);
j=1;
for i=1:6400:160000
    [N_spec(:,j:(j+99)), N_freq]= ...
        FFTanalysisMat(N_FWspeed(i:i+6399,:),20000);
    j=j+100;
end
clear j i

% There are 60 cases of overpressure, misfire and knocking. The
% corresponding final matrices will have 1500 columns

O_spec=nan(3201,60*25);
M_spec=nan(3201,60*25);
K_spec=nan(3201,60*25);
j=1;1
for i=1:6400:160000
```

```
    [O_spec(:,j:(j+59)), O_freq]= ...
        FFTanalysisMat(O_FWspeed(i:i+6399,:),20000);
    [M_spec(:,j:(j+59)), M_freq]= ...
        FFTanalysisMat(M_FWspeed(i:i+6399,:),20000);
    [K_spec(:,j:(j+59)), K_freq]= ...
        FFTanalysisMat(K_FWspeed(i:i+6399,:),20000);
    j=j+60;
end
clear j i

% Upper limmit of the frequency range
    maxF=200;

% Detect peaks in the spectrum
    [N_pks, N_frqs]=maxIsol(N_freq,N_spec,maxF);
    [K_pks, K_frqs]=maxIsol(K_freq,K_spec,maxF);
    [M_pks, M_frqs]=maxIsol(M_freq,M_spec,maxF);
    [O_pks, O_frqs]=maxIsol(O_freq,O_spec,maxF);

%% 3. Data for the Classification Learner App

peaks4analysis=[N_pks' ones(length(N_pks),1)*0
                K_pks' ones(length(K_pks),1)*1
                M_pks' ones(length(M_pks),1)*2
                O_pks' ones(length(O_pks),1)*3];


%% 4. DATA for the Neural Net Pattern Recognition app

% Dataset of samples
peaks4NN=peaks4analysis(:,1:end-1);

% Output matrix
output_matrix=zeros(7000,4);
output_matrix(1:2500,1)=1;
output_matrix(2501:4000,2)=1;
output_matrix(4001:5500,3)=1;
output_matrix(5501:end,4)=1;

%% Functions:

function [ P1, f ] = FFTanalysisMat(data, Fs)
% FFTanalysis performs a FFT analysis with matrix or array as input
% INPUTS: the data in time domain and the sampling frequency (Fs)
% OUTPUTS: the single-sided amplitude spectrum P1 and the frequency f
```

```
% Sampling period
    T = 1/Fs;

% Length of signal: how many data in the signal vector
    L = length(data);

% Compute the Fourier transform of the signal
    Y=fft(data);

% Compute the two-sided spectrum P2
    P2 = abs(Y/L);

% Compute the single-sided spectrum P1 based on P2 and
% the even-valued signal length L
    P1 = P2(1:L/2+1,:);
    P1(2:end-1,:) = 2*P1(2:end-1,:);

% Define the frequency domain f
    f = (Fs*(0:(L/2))/L)';
end


function [pks,frqs] = maxIsol(freq,amplitude,maxF,dpeaks,range)
% maxIsol finds the values of the amplitudes corresponding to half
% order of the signal
%
% INPUTS: [frequency, amplitudes, maxFreq, dpeaks, range]
%      or [frequency, amplitudes, maxFreq]
%           assuming dpeaks=6.25 Hz and range=0.25 Hz
%      or [frequency, amplitudes]
%           assuming maxFreq=200 Hz, dpeaks=6.25 Hz, range=0.25 Hz
% OUTPUTS: [value of the peaks, frequencies of the peaks]

% Setting values:
    if nargin<=3
        dpeaks=6.25;    % distance between peaks
        range=0.25;     % search interval +/-range
        if nargin<=2
            maxF=200;   % max frequency to be evaluated
        end
    end

% Create a vector with the expected peaks location
    fPeaks=(0:dpeaks:maxF)';
```

```
% Create a matrix of frequencies (repeating the same column for
%       the amount of columns present in 'amplitude')
    freq2=repmat(freq,1,size(amplitude,2));

% preallocation of variables
    pks=nan(length(fPeaks)-1,size(amplitude,2));
    frqs=nan(length(fPeaks)-1,size(amplitude,2));

% Go through the peaks and look for the global maximum within
%       the range around each peak
    for i=1:length(fPeaks)-1

        % the max in the range around the peak
        pks(i,:)=max(amplitude(freq>fPeaks(i)-range &...
            freq<fPeaks(i)+range,:),[],1);

        % save the frequencies corresponding to the peaks
        frqs(i,:)=freq2(amplitude==pks(i,:));
    end

end
```